Infrastructure and Algorithms for Information Retrieval Based On Social Network Analysis/Mining

Mohamed Reda Bouadjenek

▶ **To cite this version:**

Mohamed Reda Bouadjenek. Infrastructure and Algorithms for Information Retrieval Based On Social Network Analysis/Mining. Information Retrieval [cs.IR]. University of Paris-Saclay, Versailles, 2013. English. NNT : . tel-01885509

HAL Id: tel-01885509
https://hal.science/tel-01885509

Submitted on 2 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES

Département d'informatique
UFR de Sciences

École doctorale Sciences et Technologies de Versailles

Title

Infrastructure and Algorithms for Information Retrieval Based On Social Network Analysis/Mining.

Titre

Infrastructure et Algorithmes pour la Recherche d'Information Basés sur l'Analyse des Réseaux Sociaux.

THÈSE

présentée et soutenue publiquement le 13 Decembre 2013

pour l'obtention du

Doctorat de l'Université de Versailles Saint-Quentin-en-Yvelines

(spécialité informatique)

par

Mohamed Reda BOUADJENEK

Composition du jury

Rapporteurs :	Dr. Sihem AMER-YAHIA	Directrice de Recherche CNRS, LIG
	Pr. Patrick GALLINARI	de l'Université de PARIS 6
Examinateurs :	Pr. Esther PACITTI	de l'Université de Montpellier
	Pr. Eric GAUSSIER	de l'Université de Joseph Fourier, Grenoble I
	Johann DAIGREMONT	des Laboratoires Bell, France
Directeur de thèse :	Pr. Mokrane BOUZEGHOUB	de l'Université de Versailles SQY
Co-directeur de thèse :	Dr. Hakim HACID	des Laboratoires Bell, France

Alcatel·Lucent Bell Labs

Laboratoire PRISM

Acknowledgements

I extend my sincere gratitude and appreciation to many people who made this PhD thesis possible.

The first persons I would like to thank are my supervisors Mokrane Bouzeghoub and Hakim Hacid, who supported me and placed their trust in me. I owe you lots of gratitude for having taught me rigor and discipline during these years.

More thanks go to my thesis evaluation panel consisting of Dr Sihem Amer-Yahia and Prof Patrick Gallinari, who accepted to review my thesis, and Prof Esther Pacitti, Prof Eric Gaussier, and Mr Johann Daigremont who accepted to be part of the evaluation committee.

I also owe a lot of thanks to all my colleges of Bell Laboratories, where I spent most of my time working on my thesis. In no particular order, I would like to thank Karim Hebbar for his special advices and his sense of humor, Amyn Bennamane for his support and the interesting discussions we had, Abderrahmane Maaradji for his advices and hospitality, Loreta Maag, Ryan Skraba, Sophie Piekarec, Makram Bouzid, Leila Bennacer, Samir Ghamri-Doudane, Vincent Toubiana, Adel Saidi, Myriam Ribière, Adnan Latif, Marwen Nouri, Erick Bizouarn, and Linas Maknavicius. Merci à vous tous!

I also would like to thank all my colleagues of the PRiSM Laboratory at the University of Versailles St- Quentin-en-Yvelines. A special thanks to Ahmed Gater and Fernando Lemos with whom I shared my office during four years. I also want to thank all members of our team: Stéphane Lopes, Béatrice Finance, Zoubida Kedad, Daniela Grigory, Kim Tâm Huynh, Hanane Ouksili, Isma Sadoun, and Sofiane Abbar. I also would like to thank other members of the PRiSM laboratory for their help and support: Chantal Ducoin, Boubkeur Boudaoud, Ahmed Kharrat, Yacine Benallouche, Karim Bessaoud, Lamia Keddar, Chahinez Hamlaoui, Fatiha Amanzougarene, and Hafid Mazouz. Hope I have not forgotten anyone.

I also thank some people I met during my studies: Youcef Ammari, Abderrahmane Saadi, Sid Ali Benbelkacem, Djamel Zabchi, Touati Hakim, Azzeddine Saiah, Nacim Redam, Lotfi Abdennebi, Rami Ghorab, Kahina Gani, Redouane Chaibi, and Mehdi Ramdani.

Finally, I owe a lot of thanks to my family, who always supported me along my life: my parents, my little sister, my brother for their encouragement and support. Thanks a lot!

Abstract

Nowadays, the Web has evolved from a static Web where users were only able to consume information, to a Web where users are also able to produce information. This evolution is commonly known as Social Web or Web 2.0. Social platforms and networks are certainly the most adopted technologies in this new era. These platforms are commonly used as a means to interact with peers, exchange messages, share resources, etc. Thus, these collaborative tasks that make users more active in generating content are one of the most important factors for the increasingly growing quantity of available data. From the research perspective, this brings important and interesting challenges for many research fields.

In such a context, a mostly crucial problem is to enable users to find relevant information with respect to their interests and needs. This task is commonly referred to as Information Retrieval (IR). IR is performed every day in an obvious way over the Web, typically under a search engine. However, classic models of IR don't consider the social dimension of the Web. They model web pages as a mixture of a static homogeneous terms generated by the same creators. Then, ranking algorithms are often based on: (i) a query and document text similarity and (ii) the existing hypertext links that connect these web pages, e.g. PageRank.

Therefore, classic models of IR and even the IR paradigm should be adapted to the socialization of the Web, in order to fully leverage the social context that surround web pages and users. This thesis presents many approaches that go in this direction. In particular, three methods are introduced in this thesis:

(i) a Personalized Social Query Expansion (PSQE) framework, which achieves social and personalized expansions of a query with respect to each user, i.e. for the same query, different users will obtain different expanded queries.

(ii) a Personalized Social Document Representation (PSDR) framework that uses social information to enhance, improve and provide a personalized social representation of documents to each user.

(iii) a Social Personalized Ranking function called SoPRa, which takes into account social features that are related to users and documents.

All these approaches have the particularity of being scalable to large-scale datasets, flexible and adaptable according to the high dynamicity of social data, and efficient since they have been intensively evaluated and compared to the closest works. From a practical point of view, this thesis led to the development of an experimental social Web search engine called LAICOS that includes all the algorithms developed throughout this thesis.

Résumé

Contexte Général

Avec l'émergence du Web social, le Web a évolué d'un Web statique, où les utilisateurs étaient seulement capables de consommer de l'information, à un Web où les utilisateurs sont aussi capables de produire de l'information. Cette évolution est connue comme le Web social ou Web 2.0. Ainsi, le Web 2.0 a introduit de nouvelles libertés à l'utilisateur dans sa relation avec le Web, en lui permettant d'interagir avec d'autres utilisateurs qui ont les mêmes centres d'intérêts. Les plateformes et les réseaux sociaux (tel que MySpace, Facebook, ou LinkedIn), les plateformes de tagging collaborative (tel que CiteULike, Flickr, ou delicious), sont certainement les technologies les plus adoptées dans ce nouveau contexte. Ces plateformes permettent aux utilisateurs d'interagir, d'échanger des messages, de partager des ressources (photos et vidéos), commenter des information, créer et maintenir des profiles, interagir via des applications, etc. En plus de ces plateformes sociales dédiées à l'interaction entre les utilisateurs, les sites Web traditionnels qui sont dédiés à fournir de l'information (tel que les journaux en lignes) tendent à devenir plus sociales en fournissant des moyens aux utilisateurs pour partager, commenter, construire, et les lier des documents [AYLY09, AYHY09], e.g. via le bouton j'aime de Facebook. Ceci a aussi été facilité grâce à des initiatives tel que OpenID [1] et OpenSocial [2].

Ainsi, ces taches collaboratives permettant à l'utilisateur d'être plus actif dans la génération du contenu sont l'un des facteurs les plus importants dans l'accroissement constant des données [3]. Du point de vue de la recherche, cela pose des défis importants et intéressants pour de nombreux domaines de recherche comme: la recherche d'information, les bases de données, la fouille de données, etc., où les axes de recherche sont principalement entraîné par: (i) l'énorme quantité de données disponibles et (ii) les connaissances potentiellement utiles, latentes dans ces données.

Dans un tel contexte, l'un des problèmes les plus cruciaux est de permettre aux utilisateurs de trouver de l'information pertinente par rapport à leurs besoins. Cette tache est communément appelée Recherche d'Information (RI). Aujourd'hui, la tache de RI est réalisée quotidiennement de façon évidente sur le Web [BYRN11], typiquement en utilisant un moteur de recherche. Cependant, les modèles classiques de RI ne prennent pas en considération la dimension sociale du Web. Ils modélisent les pages Web [4] comme une mixture de homogène de termes générés par les mêmes auteurs, i.e. les auteurs des pages Web. Ensuite, les algorithmes de classement sont souvent basés sur : (i) la similarité entre les documents et les requêtes (e.g. similarité du cos-

1. http://www.openid.net/
2. http://www.opensocial.org/
3. D'autres paramètres participent à cette génération de données comme les capteurs, les périphériques connectés, etc.
4. Dans cette thèse nous faisons aussi référence au page Web comme des documents ou des ressources.

inus, Okapi BM25 [RSJ88], etc), et (ii) les liens hypertextes qui connectent ces pages Web (e.g. PageRank [BP98], HITS [Kle99], TrustRank [GGMP04], etc.).

Par conséquent, ces modèles classiques de RI et même le paradigme de RI doivent être adaptés a cette socialisation du Web, afin de tirer pleinement profit du contexte social qui entoure les pages web et les utilisateurs. En effet, exploiter l'information sociale présente de nombreux avantages (pour le RI en particulier) :

1. L'information dans les réseaux sociaux est fournie directement par l'utilisateur, ce qui fait que des informations précises concernant les centres d'intérêts de l'utilisateur peuvent être capturées.
2. De informations très récentes sont collectées sur les médias sociaux, puisque les utilisateurs expriment activement et régulièrement leurs opinions et leurs intérêts.
3. De la connaissance très précise peut être apprise sur les utilisateurs et les pages Web qui peut être réutilisé pour des services à valeur ajoutée. Par exemple, la publicité ciblée est un système basé sur les profiles, qui a démontré son efficacité dans certaines plateformes sociales comme Facebook.
4. Une énorme quantité d'informations sociales est publiée et disponible avec l'accord des éditeurs, e.g. les tweets, les annotations, les commentaires, les notes, etc.
5. Exploiter l'information sociale ne viole pas la vie privée de l'utilisateur. Le principal objectif des réseaux sociaux est de partager de l'information plutôt que de faire de la réticence de l'information.
6. Les ressources sociales sont souvent facilement accessibles, puisque la plupart des réseaux sociaux fournissent des API pour accéder à leurs données (même si souvent, un contrat de monétisation doit être établie avant toute utilisation).
7. Une structure latente sociale qui caractérise les entités d'un point de vue social peut être appris sur les entités, e.g. la pertinence sociale qui caractérise les entités du point de vue de leur intérêt.

Un des aspects les plus importants dans l'exploitation des informations sociales est le fait que l'information est fournie explicitement par les utilisateurs. Le Tableau 1 résume quelques caractéristiques des méthodes de feedback implicites classiques et des méthodes de feedback social explicites pour le RI. Les méthodes de feedback implicites décrites sont supposées être celles actuellement utilisées par les moteurs de recherche, qui reposent sur l'analyse et la fouille des logs de recherche.

En ce qui concerne le contenu d'un réseau social, il ya beaucoup d'interactions qui peuvent être exploitées pour améliorer une stratégie de RI. Le Tableau 2 illustre certaines de ces interactions. Notez que par utilisateurs, nous faisons référence à tout utilisateur d'un réseau social. Alors que pour information, nous entendons toute connaissance qui porte sur un fait ou une circonstance particulière.

Enfin, au niveau recherche, l'énorme quantité d'information patente et latente produites continuellement dans les réseaux sociaux offre une occasion unique pour

Table 1 – Feedback implicites classiques vs feedback social explicites.

Méthodes de feedback implicites classiques	**Méthodes de feedback social explicites**
Pas de participation de l'utilisateur dans le processus de feedback.	Feedback fournit directement par l'utilisateur.
Le feedback est dérivé implicitement par le système sur la base du comportement de l'utilisateur.	L'utilisateur fournit explicitement les informations de feedback.
Les profiles des utilisateurs générés ne sont pas très précis.	Des informations précises concernant les centres d'interêts des utilisateurs peut apprises.
Les données ne sont pas publiquement accessibles.	Les données sont publiquement accessibles.
Les sources de données sont souvent des courriels, logs de recherche, l'historique de navigation, etc.	Les sources de données sont souvent des commentaires, des annotations, des notes, des tweets, des étiquettes, etc.

Table 2 – Les interactions possibles qui pourraient être extraites des plateformes sociales pour améliorer le processus de RI.

Utilisateur $\leftrightarrow$ Information	**Utilisateur $\leftrightarrow$ Information**	**Utilisateur $\leftrightarrow$ Utilisateur**
- Creation - Consultation - Commenté - Description - Partage - etc...	- Confirmation - Remplacement - Enrichissement - Description - Indexation - etc...	- Amitier - Influence - Mentor - Hiérarchie - etc...

améliorer la RI. Cela a attiré l'attention de nombreux chercheurs dans la communauté de la IR, soit en proposant d 'améliorer les techniques existantes de RI [XBF+08, HKGM08, CZG+09, VCJ10, BHBV13], ou en proposant de nouveaux paradigmes de recherche [HK10, DKMS11].

Contexte de la Thèse

Cette thèse est initiée par les Laboratoires Bell, qui étudient les interactions sociales et leurs relations avec les nouvelles technologies. En particulier, le Département PPDM (Privacy Préserver, analyse de données, et de gestion) des Laboratoires Bell travaille sur un projet stratégique de communication sociale. La principale motivation derrière ce projet est de faciliter la communication entre les utilisateurs, à la fois dans les environnements physiques et numériques. A cet effet, le principal défi du

département PPDM était de construire un réseau dynamique d'utilisateurs et d'objets liés à un utilisateur particulier, concernant ses activités passées et futures. L'objectif est de fournir aux utilisateurs la proximité sociale la plus appropriée, et des services à valeur ajoutée qui s'appuient entièrement ces interactions.

Par conséquent, cette thèse fait suite aux travaux initiés par le département PPDM des Laboratoires Bell et l'équipe AMIS (Advanced Modeling of adaptive Information Systems) du laboratoire PRISM de l'Université de Versailles dans le cadre des réseaux sociaux. Notre équipe de recherche a étudié de nombreuses méthodes pour la fouille et la découverte d'information dans les réseaux sociaux, le but ultime étant de tirer pleinement parti de l'énorme quantité d'information disponible sur les médias sociaux. Les travaux de l'équipe ont mené à plusieurs contributions aux problèmes de: détection d'identité [GHS12], reconstruction de conversations [LVHAM12], la fouille de graphes sociales [BHAC11b, BHAC11a], l'agrégation de contenu social [DPK+12b], la composition sociale de services Web [MHDC10, MHSV11], l'analyse des mondes virtuels sociales [HHM+11], la personnalisation [ABK+08], et la contextualisation [ABKL09, ABL10].

Cette thèse a été en premier initiée par un projet interne chez les Laboratoires Bell, qui a ensuite abouti à de nombreuses collaborations avec des organisations universitaires et industrielles. Ces collaborations ont abouti à un projet scientifique nommé SocialSensor [DPK+12a] [5], à laquelle cette thèse a été intégrée. SocialSensor est axée sur le développement d'un nouveau Framework pour permettre l'indexation en temps réel de contenu multimédia et la recherche dans le Web social. Le projet est passé au-delà de l'indexation conventionnelle basée sur le contenu textuel en fouillant et agrégeant le contenu et l'information de l'utilisateur à travers plusieurs sites de réseautage social. L'objectif général de cette thèse est de capturer, de représenter, d'index et de rechercher des sources sociales ainsi que le Web pour fournir des informations pertinentes et des résultats sensibles au contexte pour les utilisateurs. Les principaux objectifs sont:

— Analyser les sources d'information (les réseaux sociaux, et le contenu du Web). La quantité de données et le taux de mises à jour sur des sites sociaux tels que Twitter et Facebook font qu'il est presque impossible d'analyser et indexer toutes ces données.

— Utiliser des techniques analytiques pour inclure des informations importantes dans les requêtes des utilisateurs. Il s'agit d'exprimer au mieux les besoins de l'utilisateur et fournir les informations les plus pertinentes.

— Produire de nouveaux algorithmes de recherche sociale pour la recherche personnalisée. En fait, les médias sociaux offrent une occasion unique d'offrir des services personnalisés, notamment des services de recherche personnalisés.

— Pré-traiter le contenu social et extraire des attributs caractéristiques. Le prétraitement du contenu social se rapporte à la modélisation du contexte social

5. http://www.socialsensor.eu/

qui entoure les utilisateurs et les pages Web, le principal objectif est toujours d'améliorer le processus de RI.

— Agréger et indexer à la fois les métadonnées social et le contenu textuel. Cette partie est cruciale et essentielle pour fournir un accès facile et rapide à l'information sociale.

— Fournir une recherche d'information sociale et des composants de filtrage. L'objectif est de fournir une architecture à base de composants, où chaque composant peut être facilement branché, activé et configuré pour fonctionner sur un système de recherche d'information classique.

Enfin, les problèmes abordés dans cette thèse sont résumés et présentés à la Section 2.1. Nous présentons en particulier les motivations et à un niveau élevé, les principaux défis abordés dans cette thèse du point de vue de la recherche.

Contributions

Tout en s'appuyant sur les systèmes de bookmarking sociaux comme sources d'information sociale, notre objectif dans cette thèse est de tirer parti de l'information sociale pour fournir des services de recherche d'information personnalisés. La personnalisation permet de différencier entre les individus en mettant l'accent sur leurs centres d'intérêt et leurs préférences. C'est un point clé dans la RI et sa demande ne cesse d'augmenter par de nombreux utilisateurs afin d'adapter leurs résultats [Bel08].

Les principales contributions de cette thèse sont résumées ci-dessous:

— *Un Survey sur l'état de l'art des méthodes et des algorithmes de recherche d'information sociale (RIS)*. Nous proposons de revoir certaines des contributions les plus importantes et les outils existants dans ce domaine pour comprendre les principes de la RIS telles qu'elle est actuellement formulée. Ensuite, nous proposons de classer ces contributions fondées sur une taxonomie que nous proposons, afin de structurer ce domaine de recherche. Enfin, nous proposons une analyse de certaines de ces contributions et des outils par rapport à plusieurs critères considérés comme essentiels pour la conception d'une approche de RIS.

— *Un Framework d'expansion de requêtes sociale personnalisées.* Les principaux résultats sont les suivants: (i) une approche pour l'extraction de concepts connexes de termes du graphe social d'un système de bookmarking social. Ces concepts connexes servent de base de connaissance sociale pour le processus d'expansion. (ii) Nous proposons un Framework d'expansion de requêtes social personnalisées appelé PSQE. Ce dernier fournit une expansion de requête dépendante de l'utilisateur basée sur la connaissance sociale construite. Un moteur de recherche traite ensuite la nouvelle requête étendue. (iii) Une implémentation de PSQE, qui a été utilisé pour évaluer notre méthode par rapport aux méthodes de l'état l'art.

— *Un Framework pour modéliser des représentations sociales et personnalisées des documents (PSDR)*. Ce Framework fournit à un document donné, différentes représentations sociales en fonction de chaque utilisateur, ainsi que des annotations des autres utilisateurs. PSDR a été évalué et comparé avec plusieurs approches de l'état de l'art, notamment des approches de ranking sociales et personnalisés.

— *Une fonction de classement social et personnalisé appelée Sopra*. Sopra est une fonction de classement qui s'appuie sur le contexte social qui entoure les utilisateurs et les pages Web pour estimer le degré de pertinence d'une page Web donnée par rapport à une requête émise par un utilisateur. Les performances de Sopra sont très convaincantes, car elle surpasse toute les méthode de l'état de l'art dans différents contextes sociaux.

— *Un prototype pour évaluer l'efficacité de nos techniques et algorithmes appelé Laicos*. Laicos est un moteur de recherche social qui met en ½uvre tous les algorithmes et méthodes développés tout au long de cette thèse. Laicos est open source et peut être utilisé pour développer des approches de RIS et les évaluer par rapport à d'autre méthodes existantes. Ce prototype peut aussi être utilisé pour aider les chercheurs dans l'évaluation et la comparaison de leur algorithmes grâce à son ouverture, son extensibilité et sa facilité d'utilisation.

Nos contributions aux problèmes de la recherche d'information social et personnalisée ont été publiés dans des conférences, des revues et des workshops. Le tableau 3 donne les publications correspondant aux chapitres de cette thèse.

Chapitre	**Publications**
3	([BHB16], Information Systems, 56:1 – 18, 2016.)
4	([BHBD11a], SIGIR 2011) ([BHBD11b], CORIA 2011)
5	([BHBV13], SIGIR 2013)
6	([BHB13c], SIGIR 2013) ([BHB13b], COSI 2013) ([BBHB13], ICWE 2013)
7	([BH12], WWW 2012) ([BHB13a], KDD 2013)

Table 3 – Correspondence entre les publications et les chapitres.

Organisation de la Thèse

Cette thèse est organisée en sept chapitres comme suit: Le Chapitre 2 présente les problémes de recherche abordés dans cette thèse. Ensuite, il présente le contexte nécessaire ainsi que les concepts de base utilisés tout au long de cette thèse. Le Chapitre 3 présente une analyse approfondie de l'état de l'art dans la recherche d'information sociale. Nous classons les contributions les plus importantes dans ce domaine. Ensuite, nous proposons d'analyser certaines d'entre elles par rapport à plusieurs critères considérés comme essentiels pour la conception d'une approche de

RIS. Dans le Chapitre 4, nous présentons notre approche d'expansion de requêtes en utilisant l'information sociale. Le Chapitre 5 présente notre contribution à la modélisation en recherche d'information en introduisant notre Framework PSDR. Notre fonction de ranking SoPRa est présentée et évaluée dans le Chapitre 6, et notre plateforme LAICOS est présentée dans le Chapitre 7. Enfin, nous concluons et nous présentons quelques perspectives de recherche dans le Chapitre 8.

Publications

This thesis is based on the following original articles:

1. Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, Social networks and information retrieval, how are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and platforms, Information Systems, Volume 56, March 2016, Pages 1-18, ISSN 0306-4379, http://dx.doi.org/10.1016/j.is.2015.07.008. Keywords: Information Retrieval; Social networks; Social Information Retrieval; Social search; Social recommendation

2. Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. LAICOS: An Open Source Platform for Personalized Social Web Search. In the proceeding of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pages 1446-1449, New York, NY, USA, 2013. ACM.

3. Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Athena Vakali. Using Social Annotations to Enhance Document Representation for Personalized Search. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '13, pages 1049–1052, New York, NY, USA, 2013. ACM.

4. Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. SoPRa: A New Social Personalized Ranking Function for Improving Web Search. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, SIGIR '13, pages 861–864, New York, NY, USA, 2013. ACM.

5. Mohamed Reda Bouadjenek, Amyn Bennamane, Hakim Hacid, and Mokrane Bouzeghoub. Evaluation of Personalized Social Ranking Functions of Information Retrieval. In Florian Daniel, Peter Dolog, and Qing Li, editors, Web Engineering, volume 7977 of Lecture Notes in Computer Science, pages 283–290. Springer Berlin Heidelberg, 2013.

6. Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. Personalized Documents Ranking With Social Contextualization. In 10ème Colloque sur l'Optimisation et les Systèmes d'Information COSI'2013, pages 64-75, 9-11 Juin 2013, Alger, Algérie.

7. Mohamed Reda Bouadjenek and Hakim Hacid. Laicos : A social web search engine. In WWW Panel CNRS, 2012.
8. Mohamed Reda Bouadjenek and Hakim Hacid. Using social annotations to enhance document representation for personalized search. Bell Labs Science Workshop, 2012.
9. Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. Social information retrieval : A preprocessing of queries,. 2nd Bell Labs Science Workshop, Villarceaux, France,, December 6-7 2011.
10. Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Johann Daigremont. Personalized Social Query Expansion Using Social Bookmarking Systems. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11, pages 1113–1114, New York, NY, USA, 2011. ACM.
11. Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Johann Daigremont. Une nouvelle approche d'expansion sociale de requêtes dans le web 2.0. In 8ème Conférence en Recherche d'Infomations et Applications, CORIA '11, pages 41–48, 2011.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 General Context

With the emergence of the social Web, the Web has evolved from a static Web, where users were only able to consume information, to a Web where users are also able to produce information. This evolution is commonly known as Social Web or Web 2.0. Hence, the Web 2.0 has introduced a new freedom for the user in his relation with the Web by facilitating his interactions with other users who have similar tastes or interests. Social platforms and networks (such as *MySpace*, *Facebook* and *LinkedIn*), collaborative tagging sites (like *CiteULike*, and *Flickr*), and microblogging sites (like *Twitter, Tumblr* and *Yammer)* are certainly the most adopted technologies in this new era. These platforms are commonly used as a means to interact with peers, exchange messages, share resources (photos and videos), comment on news, create and update their profiles, interact via social applications and games, etc. In addition to dedicated social platforms, traditional content providers sites like newspapers, tend to be more social since they provide to users means for sharing, commenting, constructing, and linking documents together [AYLY09, AYHY09], e.g. the Facebook like button. This has been also facilitated by initiatives like OpenID [6] and OpenSocial [7].

These collaborative tasks that make users more active in generating content are among the most important factors for the increasingly growing quantity of available online data [8]. From the research perspective, this brings important and interesting challenges for many research fields like: hardware, information retrieval, databases, data mining, etc., where the challenging dimensions are mainly driven by: (i) the huge quantity of available data and (ii) the potential hidden and useful knowledge in that data.

In such a context, a crucial problem is to enable users to find relevant information with respect to their interests and needs. This task is commonly referred to as

6. http://www.openid.net/
7. http://www.opensocial.org/
8. Other parameters participate in this phenomena like data generated by sensors, connected devices, etc.

Information Retrieval (IR). IR is performed every day in an obvious way over the Web [BYRN11], typically using a search engine. However, classic models of IR don't consider the social dimension of the Web. They model web pages [9] as a mixture of static homogeneous terms generated by the same creators, i.e. the authors of the web pages. Then, the ranking algorithms are often based on: (i) a query and document text similarity (e.g. the cosine measure, the Okapi BM25 [RSJ88], etc.), and (ii) the existing hypertext links that connect these web pages (e.g. *PageRank* [BP98], *HITS* [Kle99], *TrustRank* [GGMP04], etc.).

Therefore, classic models of IR and even the IR paradigm should be adapted to the socialization of the Web, in order to fully leverage the social context that surrounds both web pages and users. Indeed, exploiting social information has a number of advantages (for IR in particular):

1. Feedback information in Social Networks is provided directly by the user, so user's interests accurate information can be harvested.
2. Fresh information is collected on social media as people actively and regularly express their opinions and interests.
3. Accurate knowledge can be learned about users and web pages, which can be reused to build value-added services. For example, targeted advertising is an existing profile based system, which has demonstrated its effectiveness in some social platforms like *Facebook*.
4. A huge amount of social information is published and available with the agreement of the publishers, e.g. tweets, annotations, comments, ratings, etc.
5. Exploiting social information should not violate user privacy. The primary goal of social networks is to share information rather than making information reluctance.
6. Social resources are often accessible, as most of social networks provide APIs to access their data (even if often, a monetized contract must be established before any use).
7. A social latent structure that characterizes entities from a social point of view can be learned about entities, e.g. social relevance that characterizes entities from the point of view of their interest.

One of the most important aspects in exploiting social information is the fact that the feedback is provided explicitly by users. Table 1.1 summarizes some characteristics of classic implicit feedback methods and social explicit feedback methods for IR. The implicit feedback methods described are assumed to be those currently used by search engines, which rely on query logs analysis and mining.

Regarding the content of a social network, there are many interactions that can be leveraged to enhance an IR strategy. Table 1.2 illustrates some of these interactions. Note that by *Users*, we refer to any user of a social network. Whereas for information, we mean any knowledge that concerns a particular fact or circumstance.

9. In this thesis, we also refer to web pages as documents or resources.

Table 1.1 – Classic implicit feedback vs Social explicit feedback methods.

Classic implicit feedback methods	**Social explicit feedback methods**
No participation of the user in the feedback process.	Feedback information is provided directly by the user.
The feedback information is derived implicitly by the system based on the user behavior.	The user explicitly provides the feedback information.
User profiles are not so accurate.	Accurate information about user interests is gathered.
Data are not publicly available.	Data are publicly available and accessible.
Data sources are often emails, search logs, browsing history, etc.	Data sources are often comments, annotations, ratings, tweets, tags, etc.

Table 1.2 – Possible interactions that could be extracted in social platforms, which can be useful in the IR process.

Users $\leftrightarrow$ Information	**Information$\leftrightarrow$ Information**	**Users $\leftrightarrow$ Users**
- Creation - Consultation - Comment - Description - Share - etc...	- Confirmation - Replacement - Enrichment - Completion - Description - Index - etc...	- Friends - Influence - Mentor - Hierarchy - etc...

Finally, at a research level, the huge quantity of patent and latent information continually produced and embedded in Social Networks provides a unique opportunity to improve the IR. This attracted the attention of many researchers in the IR community by, either improving existing IR techniques [XBF+08, HKGM08, CZG+09, VCJ10, BHBV13], or proposing new search paradigms [HK10, DKMS11].

1.2 Thesis Context

This thesis is initiated by Bell Laboratories, which is studying social interactions and their relations with new technologies. In particular, the PPDM (Privacy Preserving, Data Analytics, and Management) department of Bell Laboratories is working on a strategic project of social communications. The main motivation behind this project is to facilitate communication between users, both in physical and digital environments. For that purpose, the main challenge of the PPDM department was to construct a dynamic network of users and objects related to a particular user, with respect

to his past and future activities. The goal is to provide to users the most appropriate social proximity, and value-added services that fully leverage these interactions.

Hence, this thesis follows the works initiated by the PPDM department of Bell Laboratories and the AMIS team (Advanced Modeling of adaptive Information Systems) of the PRiSM laboratory at Versailles University in the context of social networks. Our research team investigated many methods for mining and discovering information in social networks; the ultimate goal being to fully leverage the huge quantity of information available on social media. The team's works led to several contributions to the problems of: identity detection [GHS12], conversations reconstruction [LVHAM12], social graph mining [BHAC11b, BHAC11a], social content aggregation [DPK+12b], social services composition [MHDC10, MHSV11], social virtual word analysis [HHM+11], personalization [ABK+08], and contextualization [ABKL09, ABL10].

This thesis was first driven by an internal project at Bell Laboratories, which then yielded many collaborations with both academic and industrial organizations. These collaborations resulted into a scientific project named SocialSensor [DPK+12a] [10], to which this thesis has been integrated. SocialSensor focused on the development of a new framework for enabling real-time multimedia indexing and search in the social Web. The project moved beyond conventional text-based indexing and retrieval models by mining and aggregating user inputs and content over multiple social networking sites. Our general purpose in this thesis was to capture, represent, index, and search from social and web sources to provide relevant, and context-aware results to users. The main objectives are:

— To crawl relevant sources of information (social networks, and web content). The amount of data and the rate of updates in social sites such as Twitter and Facebook make it almost impossible to analyze and index all this data.

— To use analytic techniques for including important information in user queries. This is to best express the user's needs and provide the most relevant information.

— To produce novel social search algorithms for personalized search. Actually, social media provide a unique opportunity to provide personalized services, in particular personalized search services.

— To pre-process social content and extract features. Pre-processing of social content refers to modeling the social context that surrounds users and web pages, the primary goal is still to improve the IR process.

— To aggregate and index both social metadata and textual content. This part is crucial and essential to provide an easy and fast access to social information.

— To provide social information retrieval and filtering components. The goal is to provide a component-based architecture, where each component can be easily

10. http://www.socialsensor.eu/

plugged, activated and configured to work on a classic information retrieval system.

Finally, the problems addressed in this thesis are summarized and presented in Section 2.1. Especially, we present the motivations and at a high level, the main challenges addressed in this thesis from a research point of view.

1.3 Main Contributions

While relying on social bookmarking systems as sources of social information, our goal in this thesis is to leverage social information to provide personalized Web search services. Personalization allows differentiating between individuals by emphasizing on their specific domains of interest and their preferences. It is a key point in IR and its demand is constantly increasing by numerous users for adapting their results [Bel08].

The main contributions of this thesis are summarized in the following:

— *A survey on different state of the art methods and algorithms of Social Information Retrieval (SIR).* We propose to review some of the most important contributions and existing tools in this area to understand the principles of the SIR as they are currently formulated. Then, we propose to categorize these contributions based on a taxonomy that we propose to structure this huge area. Finally, we propose an analysis of some of these contributions and tools with respect to several criteria considered as crucial for designing an effective SIR approach.

— *A Personalized Social Query Expansion framework.* The main results are: (i) an approach for extracting related concepts of terms from the social graph of a social bookmarking system. These related concepts serve as social knowledge for the expansion process. (ii) We propose a Personalized Social Query Expansion framework called *PSQE*. This latter provides a user-dependent query expansion based on the constructed social knowledge. A search engine then processes the resulted expanded query. (iii) An implementation of the *PSQE* framework, which was used for evaluating our method with respect to the closest state of the art methods.

— *A Personalized Social Document Representation framework (PSDR).* This framework is expected to deliver for a given document, different social representations according to each user, based on the feedback of other users. The PSDR approach has been evaluated with respect to many state of the art approaches including personalized social ranking approaches.

— *A Social Personalized Ranking function called SoPRa.* SoPRa is a ranking function that leverages the social context that surrounds users and web pages for estimating the degree of relevance of a given web page with respect to a query issued by a user. The performances of SoPRa are very convincing since it outperforms the state of the art methods in different social contexts.

— *A prototype for evaluating the effectiveness of our algorithms and techniques called LAICOS*. *LAICOS* is a Social Web search engine that implements all the algorithms and methods developed throughout this thesis. As described later, *LAICOS* is an open source Social Web search engine, which can be used for developing SIR approaches and evaluating them against the closest state of the art methods. It is expected to help researchers in the evaluation and comparison tasks thanks to its openness, its extensibility, and its ease of use.

Our contributions to the problems of personalized Social Web search have been published in conferences, journals and workshops. Table 1.3 gives the publications corresponding to the thesis chapters.

Chapter	**Publications**
3	([BHB16], Information Systems, 56:1 – 18, 2016.)
4	([BHBD11a], SIGIR 2011) ([BHBD11b], CORIA 2011)
5	([BHBV13], SIGIR 2013)
6	([BHB13c], SIGIR 2013) ([BHB13b], COSI 2013) ([BBHB13], ICWE 2013)
7	([BH12], WWW 2012) ([BHB13a], KDD 2013)

Table 1.3 – Correspondence between publications and chapters.

1.4 Thesis Organization

The remaining of this thesis is organized in seven chapters:

Chapter 2 presents the research issues tackled in this thesis from a high level point of view. Then, it introduces the background needed as well as the basic concepts used throughout this thesis, including Information Retrieval and social networks.

Chapter 3 presents a deep analysis of the state of the art in Social Information Retrieval. We categorize the most important contributions in this area. Then, we propose to analyze some of them with respect to several criteria considered as crucial for designing an effective SIR approach.

Chapter 4 presents our proposal for query expansion using social information. We propose an approach that considers: (i) the semantic similarity between tags composing a query, (ii) a social proximity between the query and the user for a personalized expansion, and (iii) a strategy for expanding, on the fly, user queries.

Chapter 5 presents our contribution to IR modeling. Since each user has his own understanding and point of view of a given document, we propose a Personalized Social Document Representation (*PSDR*) of each document per user based on his social activities. The proposed approach relies on matrix factorization to compute the PSDR of documents that potentially match the query at query time. The complexity analysis shows that our approach scales linearly with the number of documents that potentially match the query and can thus be applied to very large datasets.

Chapter 6 presents our contribution to IR modeling by studying personalized ranking functions.

We first propose a deep analysis of the state of the art personalized social ranking functions. This analysis includes a discussion on the effectiveness, the weakness and the performance of each approach in different contexts.

Then, based on the technical issues identified in the previous personalized social ranking functions, we propose SoPRa, a new social ranking function. SoPRa is expected to handle these issues while considering the social dimension of the Web. This social dimension is any social information that surrounds documents along with the social context of users. Currently, SoPRa relies on folksonomies for extracting these social contexts, but it can be extended to use any social meta-data, e.g. comments, ratings, tweets, etc. SoPRa has been evaluated by an offline study and by a user survey over a large public dataset of *delicious*, and compared to the closest state of the art methods. The obtained results show significant benefits for personalized search.

Chapter 7 introduces *LAICOS*, a Social Web search engine as a contribution to the growing area of social information retrieval (SIR). Social information and personalization are at the heart of *LAICOS*. On the one hand, the social context of documents is added as a layer to the textual content traditionally used to index a collection of documents to provide a personalized social representation of documents using the PSDR framework. On the other hand, the social context of users is used for profiling users, and providing personalized search results through the Personalized Social Query Expansion framework (PSQE) and the Social Personalized Ranking function (SoPRa). We describe the different components of the system, while relying on social bookmarking systems as a source of social information for personalizing and enhancing the IR process. We show how the internal structure of indexes as well as the query expansion process operated using social information.

Chapter 8 presents a conclusion and research perspectives.

Chapter 2

Research Issues and Background

This chapter reviews and analyzes the main concepts addressed and required in this thesis. In Section 2.1, we present the motivations and, at a high level, the main challenges addressed in this thesis from a research point of view. Then, in Section 2.2, we present the background knowledge required in this thesis. Especially, we introduce information retrieval and its basic process, social networks, and the main models of social relationships. In Section 2.3 we introduce social information retrieval, which is defined as the bridge that fills the gap between information retrieval and the Web 2.0. The notation used throughout this thesis is summarized in Section 2.4. Finally, we will conclude this chapter in Section 2.5.

2.1 Motivation and Research Issues

Finding relevant information on the Web remains challenging in many cases for end-users as: (i) usually, the user doesn't necessarily know what he is looking for until he reaches it, and (ii) even if the user knows what he is looking for, he doesn't always know how to formulate the right query to find it (except if in cases of navigational queries [Bro02]).

In the following, we first start by providing two examples of problems that demonstrate the richness of social information embedded in social networks, that we will exploit in the solution approaches introduced in this thesis. Then, we will summarize the research issues that emerge from the analysis of these two motivating examples.

2.1.1 Motivating Examples

The following examples are certainly classical examples in information retrieval, but we believe they are useful for explaining the challenges we are trying to tackle and the motivation behind this thesis.

Example 2.1. Alice is a novice user in computer science tools but familiar with the Web and Web 2.0 platforms. She is mainly a fan of travels but currently she is looking

to install an operating system on her new acquired machine. She heard about an operating system called "Linux" and she starts searching for "Linux" using her favorite search engine (based on keywords matching). In the meanwhile, Alice has absolutely no idea about the different distributions of Linux she could install, e.g. "Ubuntu", "Fedora", "Debian", etc. When she searches using the keyword "Linux", the search engine returns all Web pages that are indexed using this keyword. Thus, Alice has to choose which distribution of Linux she has to install, even if she doesn't know which one to choose.

From this situation, we can extract two main problems:

(i) From the user perspective, Alice has to know which distribution of Linux she needs, to be able to decide which one to install.

(ii) From the system perspective, documents that are not indexed with the keyword "Linux", but rather with Ubuntu or Debian are relevant to this query but are not retrieved.

As for the first problem (i), if we suppose that Oscar, a friend of Alice on Facebook has a strong background in computer science, he can help Alice to choose the right distribution to install. Therefore, we believe that a search engine should leverage such information during query processing, and should propose to connect Alice to Oscar. Eventually, Oscar could help and assist Alice in her IR task in order to make the right choice.

Regarding the second problem (ii), traditional IR strategies, which are mainly based on a matching between the query terms and the index terms, are not able to retrieve relevant documents, which are not indexed with terms of the query. However, if we consider the social context that surrounds documents, many social metadata are associated to them (metadata can be comments, annotations, tweets, etc.). These social metadata describe documents from the social community point of view. Eventually, the relevant documents, which are not indexed with query terms, may be rather described with these terms by the social community. Hence, these social metadata are highly valuable and should be considered by search engines to improve the results quality.

Example 2.2. Bob, a friend of Alice, is also a novice in computer science and he is looking for information to plan his holidays to Java in the Pacific Ocean. When he typed the query "java" on his favorite search engine, he was directly offered huge lists of documents talking about a programming language called "Java". The retrieved documents are not really relevant considering the expectations of Bob.

This situation clearly shows two problems: (i) an ambiguity problem, where the search engine fails to discriminate between the expectations of the user and the priority it gives to the results, and (ii) the query formulation problem, where the user needs to provide additional effort to expect more precise results. These problems can also be addressed by considering the social proximity the user has with his surrounding

social relatives. This is by considering that Bob hasn't shown any interest in computer science on his social networks, and the fact that he is strongly related to people who are fan of travels like Alice.

The different situations described above show some of the challenging issues in information retrieval [11]. However, in all these situations, social information represents a valuable source of knowledge, which can be reused to improve the IR process in order to provide suitable results and tackle the described problems.

2.1.2 Research Issues

Leveraging the social information into information retrieval involves a certain number of technical issues addressed in this PhD thesis:

1. Leveraging social information in the IR process. This is certainly the main goal of this thesis, and still the ultimate goal to achieve. As it has been shown above, social information represents a valuable source of knowledge, which can be reused to improve many services (especially search services).
2. Providing a certain personalization in the relevance of search results. As indicated in [PSC+02], relevance is relative for each user.
3. Providing mechanisms to assist users in formulating their queries. Even if such a mechanism is in back-office, an IR system should be aware about the social relative that surround users in order to properly reformulating their queries.
4. Modeling documents while considering their social contexts. Here, modeling refers to the definition of a conceptual model for representing documents. Eventually these representations should include this social context.
5. Modeling users' profiles by considering their social context. Social information can be a highly valuable and trusted source of information that can be used to model users' profiles. This is due to the fact that feedback information is provided explicitly by users.
6. The ability to leverage any source of social information in the IR process. Indeed, users are actually connected to many social platforms, e.g. Facebook, Tweeter, etc., and they reveal part of their interests in each platform. Hence, aggregating social information from different social platforms is a key problem in the context of this PhD thesis.

In this thesis we address these issues as follows: Chapter 4 addresses issues 1, 2, 3, and 5. Chapter 5 addresses issues 1, 2, 4, and 5. Chapter 6 addresses issues 1, 2, 4 and 5. Finally, Chapter 7 addresses issues 6.

11. Other motivating examples are given by Amer-Yahia et al. in [AYLY09].

2.2 Background

This section describes and defines the basic concepts used throughout this thesis.

2.2.1 Information Retrieval

Information Retrieval is a process of recovering stored information from large datasets to satisfy user's informational needs. Salton [Sal68] and Baeza et al. [BYRN11] defined IR as follows:

Definition 2.1. [Information Retrieval] Information Retrieval (IR) is the science that deals with the representation, storage, organization of, and access to information items in order to satisfy the user requirements concerning this information.

Although the characterization of user's needs is not a simple task, users generally specify their requirements under the form of queries that the IR system should process to determine and present the documents that match their needs. Google, Bing and Yahoo! are certainly the most known information retrieval systems. In such systems, users express their requirements in the form of keywords, which are generally considered as a summary of the user's information needs. Given a query, the IR system attempts, following a set of processes, to retrieve information which may be relevant to users. An IR system is evaluated in its accuracy and ability to retrieve high quality information/documents, which maximize users' satisfaction, i.e. the more the answers correspond to the users' expectations, the better the system is.

From an architectural point of view, the IR process is composed mainly of the following two complementary sub-processes:

1. An off-line sub-process illustrated in the right part of Figure 2.1. The document collection is crawled and browsed in order to retrieve all documents through the potential ties that link these documents together. For each retrieved document, a processing is applied consisting mainly in reducing its full set of words to a set of index terms. A specific organization of the whole extracted content is applied to create an index over the collection of documents. The periodicity of the crawling execution and updates of indexes depends on the policy adopted by the data extraction engine.
2. An on-line sub-process illustrated in the left part of Figure 2.1 that takes in charge the user's query. The query is sent generally under the form of keywords and is reduced by the query-processing engine following the same strategy as that of documents processing/indexing. The resulted set of terms of the user query is often enriched [Eft96] (expanded) or refined [KC09] (by removing some terms). Then, the query is further processed to obtain a set of documents using the index structure previously built. This list is composed of documents that are related to the query terms. Next, the retrieved documents are ranked according to a likelihood of relevance to the query and the user from the more relevant

to the less relevant one. This is the most critical step because the quality of the results, as perceived by the users, is fundamentally dependent on the ranking. Finally, the top ranked documents are then formatted for presentation to the user.

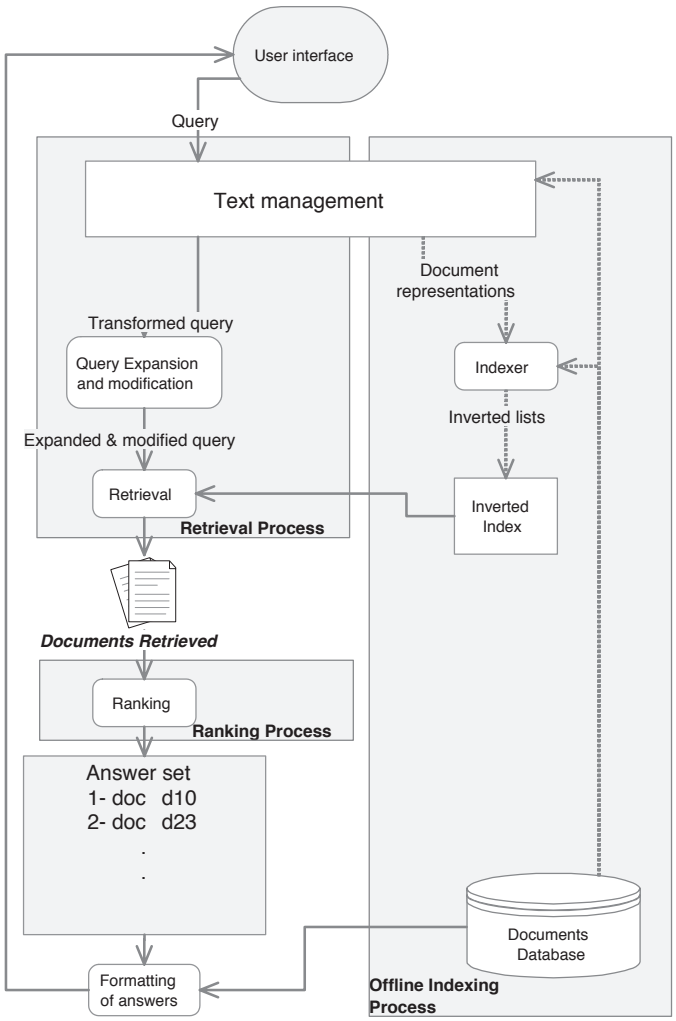


Figure 2.1 – The process of indexing, retrieval, and ranking of documents as described in [BYRN11].

These sub-processes have been improved by adding several features to deal with very large corpora, e.g. the Web, in which there might be a millions of pages that match a query. In this context, effectively ranking those pages is a key problem. Hence, several algorithms have been proposed to statistically estimate the relevance of web pages. The basic idea is to merge the ranking score of a document with respect to a query with its estimated popularity (or quality) value. This aims at highlighting and pushing up documents that are popular and attract much attention (documents with higher popularity). Among these algorithms we cite: *HITS* [Kle99], *TrustRank* [GGMP04] and *PageRank* [BP98]. This latter is probably the most popular

one, and is part of the ranking algorithm originally used by Google. *PageRank* considers the number of hyperlinks that point to a page as well as the number of hyperlinks that this page contains. Briefly, PageRank simulates the behavior of a user who will randomly click on links of a Web page.

Example 2.3. [Google PageRank] If a user Alice is at a page a, she will randomly move to one of the pages pointed by a by clicking on the considering hyperlink. Next, she repeats the process for the page she moved to, and so on. After a large number of such moves, we can compute the probability with which Alice visited each page. This probability is referred to the PageRank of that Web page.

2.2.2 IR Model

Modeling in IR consists of the definition of a conceptual model for representing documents and queries. Many IR models have been proposed among which: the boolean model, the vector space model, and the probabilistic model [RJ76]. These IR models are well described in [BYRN11]. In this PhD thesis we mainly rely on the Vector Space Model (VSM) for its simplicity and its wide usage (especially because it is used in the Lucene search engine [12]).

Briefly, in VSM, resources (documents, users' profiles, etc.), and queries are mapped to vectors and are represented as t-dimensional vectors given by:

$$q = (w_{1,q}, w_{1,q}, \cdots, w_{t,q}),\ r_j = (w_{1,j}, w_{1,j}, \cdots, w_{t,j}) \tag{2.1}$$

where $w_{i,k}$ is the weight associated to a particular vector, with $w_{i,k} \geq 0$. We choose to use the TF-IDF measure to compute this weight and the cosine similarity to compute similarity between these vectors as follows:

$$TF - IDF_{t_i,r_j} = TF_{t_i} \times log\left(\frac{|R|}{|R_{t_i}|}\right), \tag{2.2}$$

$$Sim(\vec{q}, \vec{r}_j) = \frac{\vec{q} \times \vec{r}_j}{\|\vec{q}\| \times \|\vec{r}_j\|} = \frac{\sum_{i=1}^{n} w_{1,q} \times w_{1,j}}{\sum_{i=1}^{n}(w_{1,q})^2 \times \sum_{i=1}^{n}(w_{1,j})^2} \tag{2.3}$$

where TF_{t_i} denotes the term frequency of t_i in the resource r_j, $|R|$ denotes the total number of resources in the whole collection and $|R_{t_i}|$ denotes the number of resources in which the term appears. Notice that the cosine similarity is particularly used in positive space, where the outcome is neatly bounded in $[0, 1]$. Hence, if two vectors have the same orientation and are equal, we have a cosine similarity of 1. However, if the two vectors are diametrically opposed we have a similarity of 0.

12. http://lucene.apache.org/core/

2.2.3 Evaluation of the retrieval quality of IR systems

This section is partially a summary of Chapter 4 of the book Modern Information Retrieval [BYRN11]. A proper definition is given regarding the evaluation of IR algorithms and systems.

Definition 2.2. *Retrieval evaluation* is a process of systematically associating a quantitative metric to the results produced by an IR system in response to a set of user queries. This metric should be directly associated with the relevance of the results to the user. A common approach to compute such a metric is to compare the results produced by the system with results suggested by humans for that same set of queries.

Notice that *Retrieval evaluation* here means evaluating the quality of the results, not the performance of the system (in term of how fast it processes queries).

Many different metrics for evaluating the retrieval quality of IR systems and algorithms have been proposed, i.e. the quality of the results. These metrics require a collection of documents and queries. All common measures described here assume a ground truth notion of relevancy: every document is known to be either relevant or non-relevant to a particular query. In practice, queries may be ill-posed and there may be different shades of relevance. In the following, we define five retrieval evaluation metrics used throughout this thesis.

2.2.3.1 Precision and Recall

Consider an information request q (of a test reference collection) and its set D_R of relevant documents. Let $|D_R|$ be the number of documents in this set. Assume that a given retrieval algorithm (which is under evaluation) process the information request q and generates a set of answers D_A. Let $|D_A|$ be the number of documents in this set. Further, let $|D_R \cap D_A|$ be the number of documents in the intersection of the sets D_R and D_A. Then the precision and recall measures are defined as follows:

— Precision is the fraction of the retrieved documents (the set D_A), which is relevant

$$Precision = p = \frac{|D_R \cap D_A|}{|D_A|}$$

— Recall is the fraction of the relevant documents (the set D_R), which has been retrieved

$$Recall = r = \frac{|D_R \cap D_A|}{|D_R|}$$

2.2.3.2 Mean Average Precision (MAP)

The main idea of the Mean Average Precision is to generate a single value summary of the ranking by averaging the precision figures obtained after each new relevant documents is observed.

Let D_{R_i} refer to the set of relevant documents for query q_i, as before, and $|D_{R_i}|$ refer to its size, i.e. the total number of relevant documents for query q_i. Let $D_{R_i}[k]$ be a reference to the k-th document in D_{R_i}. Then, $P(D_{R_i}[k])$ is the precision when the $D_{R_i}[k]$ document is observed in the ranking of q_i. If that document is never retrieved, a common occurrence in real searches, $P(D_{R_i}[k])$ is taken as zero (it is actually undefined but we can assume that it is small and approximate it to zero). The average precision AP_i for query q_i is defined as follows:

$$AP_i = \frac{1}{|D_{R_i}|} \sum_{k=1}^{|D_{R_i}|} P(D_{R_i}[k])$$

MAP, the Mean Average Precision over a set of queries is then defined as follows:

$$MAP = \frac{1}{|Q|} \sum_{i=1}^{|N_q|} AP_i \tag{2.4}$$

where $|Q|$ is the total number of queries.

2.2.3.3 Mean Reciprocal Rank (MRR)

There are instance in which we are interested in the first correct answer to a given query task. For example URL and homepage Web queries, in which the users specify a URL or a reference to a homepage and are interested in the first correct answer. In these instance, Mean Reciprocal Rank is a metric that favors results whose first correct answer is higher in the ranking is preferred.

Let D_{A_i} be the ranking relative to a query q_i and let $S_{correct}(D_{A_i})$ be a function that returns the position of the first correct answer D_{A_i}. Given a threshold ranking position S_h, the reciprocal rank of D_{A_i} is defined as

$$\begin{cases} \frac{1}{S_{correct}(D_{A_i})} & if\ S_{correct}(D_{A_i}) \leq S_h \\ 0 & otherwise \end{cases}$$

That is, the reciprocal rank is zero if the first correct results occurs at a position in the ranking beyond S_h. For a set Q of queries, the Mean Reciprocal Rank (MRR) is the average of all reciprocal ranks, which is computed as follows:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{q} \frac{1}{S_{correct}(D_{A_i})} \tag{2.5}$$

MRR is a metric that favors ranking whose first correct results occur near the top.

2.2.3.4 Discounted Cumulative Gain (nDCG)

Precision and recall, while broadly used, allow only binary relevance assessments and might be heavily influenced by outliers, i.e. relevant documents found late in

the ranking. As a result, they might blur the distinction between an IR model that retrieves highly relevant documents at the top of the ranking and another model that retrieves only mildly relevant documents at the top of the ranking. These limitations can be overcome by adopting graded relevance assessments and metrics that combine them effectively, such as Discounted Cumulated Gain (DCG).

The DCG accumulated at a particular rank position p is defined as:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{2.6}$$

Since result set may vary in size among different queries or systems, to compare performances the normalized version of DCG uses an ideal DCG. To this end, it sorts documents of a result list by relevance, producing an ideal DCG at position p ($IDCG_p$), which normalizes the score:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{2.7}$$

In the next section, we discuss some basic aspects in the area of Social Networks Analysis, a valuable input to understand the underlying principles of social information retrieval [13].

2.2.4 Social Networks and SNA

Nowadays, social networks are at the heart of the Web 2.0. A Social Network is defined as follows:

Definition 2.3. [Social Network] A social network is the social structure, which emerges from human interactions through a networked application.

Example 2.4. [Online Social Network] *Facebook* is certainly the most popular social network that handles relationships between individuals. There are many social network sites that manage in addition to the users, objects. These include documents on *CiteULike*, images on *Flickr*, videos on *YouTube*, etc.

The structure of a social network can be constructed in two ways: (i) either explicitly declared by the user, e.g. friendship links in *Facebook*, or (ii) implicitly inferred from the behavior and the common interests of users, e.g. Social Network of Web services [MHDC10, MHSV11]. To understand the underlying social structures and phenomena, a set of techniques and methods exist, which are known as Social Networks Analysis (SNA) techniques [SK94]. SNA introduces methods and metrics (e.g. centrality, influence, etc) for analyzing a Social Network, e.g. measuring the role of individuals and groups of individuals in a Social Network.

13. Note that we don't intend to provide a complete review of the SNA field but rather to provide the main underlying principles, which could be leveraged in IR systems and which are helpful to understand the analysis provided in this paper.

Each social network might be characterized by the relationships that link its users, e.g. friend relationships, follower-following relationships, publisher-subscriber relationships, etc. Hence, we distinguish mainly three models of social relationships that we describe hereafter [14].

2.2.4.1 Symmetric Social Network Relationships (SSN)

Many social networks manage symmetric relationships that translate the same consideration of relations between entities, i.e. users, participating in the relation. Social networks that include these relationships allow users to maintain a list of friends and thus create friendship relations. The *Friendship* relation is of the form Alice considers Bob as a friend and she explicitly adds Bob to her list of friends. This relation is instantiated once Bob accepts the request. Thus, the friendship strength of a link between two users can reflect for example, the degree of trustiness, the degree of mutual interest, etc. This model is more dedicated to create and maintain personal relationships with trusted persons and with whom one is expected to share personal elements and contents. Formally, the *symmetric relationship* can be defined as follows:

Definition 2.4. [Symmetric Relationship] The relationship $\mathcal{R}$ over a set of user U is symmetric if: $\forall u_1, u_2 \in U, u_1 \mathcal{R} u_2 \Rightarrow u_2 \mathcal{R} u_1$.

Example 2.5. Facebook is a good example of social network that includes these relationships. Let consider the users *Alice* and *Bob* on *Facebook*, if *Alice* has a relationship with *Bob* then *Bob* has the same relationship with *Alice*.

2.2.4.2 Asymmetric Social Network Relationships (ASN)

Similarly, many social networks manage social relationships that can link two users with two different perspectives depending on the user. These relationships illustrate the concept of followers-following or publisher-subscriber and are generally at the heart of microblogging platforms, e.g. *Twitter* and *Yammer*. These social networks allow a user to create and maintain a list of following people by permitting him to subscribe to their information stream. This model of social networks is more dedicated to the dissemination of information than to mutual sharing of information. Formally, the *asymmetric relationship* can be defined as follows:

Definition 2.5. [Asymmetric Relationship] The relationship $\mathcal{R}$ over a set of user U is asymmetric if: $\exists u_1, u_2 \in U, u_1 \mathcal{R} u_2 \Rightarrow \neg(u_2 \mathcal{R} u_1)$.

Example 2.6. Consider a relationship between a user *Alice* and a user *Bob* on *Twitter*. *Alice* may subscribe to the content published by *Bob*, while *Bob* does not necessarily subscribe to the content of *Alice* as he considers her from his perspective.

14. We only describe the main social relationships models that we believe are the most adopted in social networks.

2.2.4.3 Ternary Relationship Social Network (TRSN)

The two previous models of social relationships involve generally only one type of nodes, i.e. users. In this last model, social relationships imply three types of nodes, i.e. users, resources, and tags. Social bookmarking websites are representatives of such models.

Social bookmarking websites are based on the techniques of *social tagging or collaborative tagging*. The principle behind social bookmarking platforms is to provide the user with means to annotate resources on the Web, e.g. URIs in *delicious*, videos in *youtube*, images in *flickr*, or academic papers in *CiteULike*. These annotations (also called tags) can be shared with others. This unstructured (or free structured) approach to classification with users assigning their own labels is often referred to as a *folksonomy [HHLS05, LHHF05].*

A folksonomy is based on the notion of bookmark, which is formally defined as follows:

Definition 2.6. [Bookmark] Let U, T, R be respectively the set of Users, Tags and Resources. A *bookmark* is a triplet *(u,t,r)* such as $u \in U, t \in T, r \in R$, which represents the fact that the user u has annotated the resource r with the tag t.

Then, a folksonomy is formally defined as follows:

Definition 2.7. [Folksonomy] Let U, T, R be respectively the set of Users, Tags and Resources. A folksonomy $\mathbb{F}(U, T, R)$ is a subset of the cartesian product $U \times T \times R$ such that each triple $(u, t, r) \in \mathbb{F}$ is a *bookmark*.

Finally, it is important to notice that most of the existing Social Networking platforms are not restricted to manage only one kind of social graph but they may manage several social relationships. For example, *delicious* which is a tagging based platform, manages:

(i) a social graph of ternary relationship between users, tags, and documents;

(ii) a social graph of publisher-subscriber to provide to users,with means to view all the bookmarks saved by interesting people, such as friends, co-workers, and favorite bloggers.

2.2.5 Representation of a Social Network

Social network representation facilitates quantitative or qualitative analysis by describing features of a network, either through numerical or visual representation. Hence, visual representations of social networks are important to understand network data and highlight the result of the analysis.

In most of the time, social networks are represented using graphs. Hence, we use the following graph representations to represent the three previous social network relationships:

— A SSN can be represented using an undirected graph $\mathbb{G} = (V, E)$ comprising a set V of vertices, e.g. users, object, etc., and a set $E \subset V \times V$ of edges where: $\forall v_i, v_j \in V$, if $(v_i, v_j) \in E$ then $(v_i, v_j) = (v_j, v_i)$, i.e. the edge (v_i, v_j) is equivalent to (v_j, v_i) as depicted in Figure 2.2.

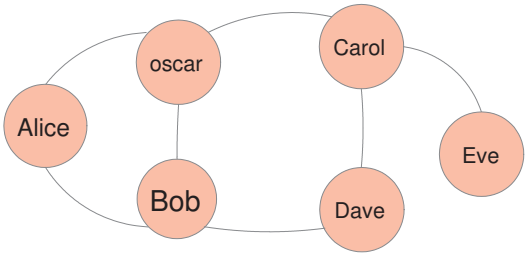


Figure 2.2 – A simple undirected social graph, that depicts the notion of friends, e.g. *Facebook*, *Orkut*.

— An ASN can be represented using a directed graph $\mathbb{G} = (V, A)$ comprising a set V of vertices, e.g. users, pages, etc., and a set $A \subset V \times V$ of arcs (directed edges) such as: $\forall v_i, v_j \in V$, if $(v_i, v_j) \in A$ then the assertion $(v_i, v_j) = (v_j, v_i)$ is not true, i.e. the edge (v_i, v_j) is not equivalent to (v_j, v_i) as depicted in Figure 2.3.

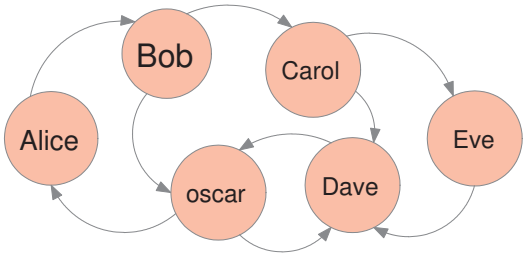


Figure 2.3 – A directed social graph, that depicts the notion of followers and following, e.g. *Twitter*, *Yammer*.

— In TRSN, a folksonomy can then be represented by a tripartite-graph where each ternary edge represents a bookmarks. In particular, the graph representation of the folksonomy $\mathbb{F}$ is defined as a tripartite graph $\mathcal{G}(V, E)$ where $V = U \cup T \cup R$ and $E = \{(u, t, r) | (u, t, r) \in \mathbb{F}\}$. Figure 2.4 displays seven bookmarks provided by two users on three resources using three tags.

2.2.6 User Modeling

Social networks have proven to be a valuable knowledge for user profiling [CZG+09, NM07, VCJ10, XBF+08]. Especially, because users comment, tag, and share interesting and relevant information to them with keywords that may be a good summary of

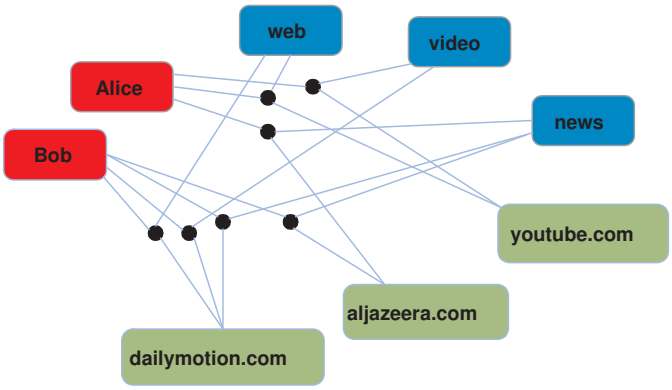


Figure 2.4 – Example folksonomy. Two users (*Alice* and *Bob*) annotate three resources (youtube.com, dailymotion.com, aljazeera.com) using three tags (news, Web, video). The triples (u, r, t) are represented as hyper-edges connecting a user, a resource and a tag. The 7 triples correspond to the following 4 posts: (*Alice*, aljazeera.com, {news}), (*Alice*, youtube.com, {Web, video}), (*Bob*, aljazeera.com, {news}), (*Bob*, dailymotion.com, {news, Web, video}).

their interest. Besides the information that a user explicitly provides regarding his interests, classic text processing techniques can be applied on user's tweets, comments, and annotations in order to extract keywords and concepts that may summarize his interests. Hence, in this thesis, the profile of a user includes all the terms he used in his social interactions along with their weights to capture user's interests. It is formally defined as follows:

Definition 2.8. [User Profile] Let U, T be respectively the set of Users and Tags (or terms). A profile assigned to a user $u \in U$, is modeled as a weighted vector $\overrightarrow{p_u}$ of m dimensions, where each dimension represents a tag the user employed in his social interactions. More formally, $\overrightarrow{p_u} = \{w_{t_1}, w_{t_2}, ..., w_{t_m}\}$ such that w_{t_m} is the weight of t_m.

Details on how we compute the weight of each dimension is given in Section 4.3.1.4.

In the next section, we discuss the concept of social information retrieval, which links IR and social networks.

2.3 Social Information Retrieval

With the emergence of the social Web, a large range of applications and services make the user becoming more interactive with Web resources, and a lot of information that concerns both users and resources is constantly generated. This information can be very useful in information retrieval tasks for both user and resource modeling, where classic models are blind to this social context.

Therefore, the fields of IR and SNA have been bridged resulting in *Social Information Retrieval (SIR)* models [GF07]. These models, in most of the time, extend conven-

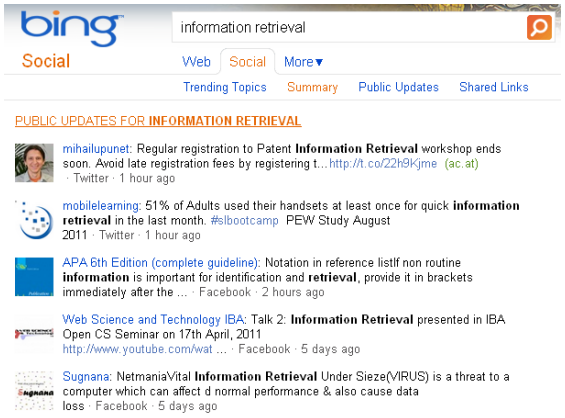
tional IR models to incorporate social information. The meaning of the concept *Social Information Retrieval* can be very broad, but we propose the following definition:

Definition 2.9. [Social Information Retrieval] *Social Information Retrieval* is the process of leveraging social information (both social relationships and the social content), extracted from social platforms, to perform an IR task with the objective of better meeting users' needs.

SIR aims at providing relevant content and information to users in the areas of information retrieval, and research; covering topics such as social tagging, collaborative querying, social network analysis, subjective relevance judgments, Q&A systems, and collaborative filtering [GF07].

Several existing platforms investigate this track in order to improve the search paradigm as illustrated in Figures 2.5 and 3.2. These platforms include *Social Bing*, *Google+*, *Aardvark* [15], *Yahoo! Answers*, etc. The research in this field has emerged and became very present in the daily (virtual) life of users. Investigating the IR field from this perspective seems to be very promising to improve the representation, the storage, the organization, and the access to information.

(a) Google Search where the button +1 has been add in order to share recommendations with friends, contacts and the rest of the Web when advice is most helpful on Google search.



(b) Bing social, a new function that has been integrated to Bing Web search which uses the content of both Twitter and Facebook as data source for real-time search.

Figure 2.5 – Example of SIR engines.

2.4 Notation

Throughout this thesis, we use the notation summarized in Table 2.1.

15. Aardvark has been acquired by Google on February 11, 2010. In September 2011, Google announced it would discontinue a number of its products, including Aardvark.

Table 2.1 – Thesis' notation overview.

Variable	Description
u, d, t	Respectively a user u, a document d and a tag or a term t.
U, D, T	Respectively a set of users, documents and tags (or terms).
$\mid A \mid$	The number of element in the set A.
T_u, T_d, $T_{u,d}$	Respectively the set of tags used by u, tags used to annotate (or describe) d, and tags used by u to annotate d.
D_u, D_t, $D_{u,t}$	Respectively the set of documents tagged by u, documents tagged with t, and documents tagged by u with t.
U_t, U_d, $U_{t,d}$	Respectively the set of users that use t, users that annotate d, and users that used t to annotate d.
$M^d_{U,T}$	The Users-Tags matrix associated to the document d.
M^d_U, M^d_T	Respectively the user latent feature matrix, and the tag latent feature matrix associated to a document d.
$\overrightarrow{p_u}$	The weighted vector of the profile of the user u.
w_i	The weight of the i^{th} dimension into a vector.
$Cos(\vec{A}, \vec{B})$	The cosine similarity measure between two vectors.
$\| . \|_F$	The Frobenius norm where: $\|M\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} \mid a_{ij} \mid^2}$

2.5 Conclusion

In this chapter, we presented the main challenges addressed in this thesis and the fundamental concepts we are using. We introduced the notion of: (i) Information Retrieval by presenting its basic process, (ii) Social Networks by presenting and defining the main models of social relationships, (iii) the way we model users in social networks, and (iii) Social Information Retrieval by defining it as the concept that bridges the gap between IR and Social Networks.

The next chapter is devoted to the state of the art. We discuss a taxonomy of SIR models regarding the way social information is used and exploited for IR tasks.

Chapter 3

Social Networks in Information Retrieval: State of the Art

This chapter presents a deep analysis of the state of the art in Social Information Retrieval. We categorize the most important contributions in this area. Then, we propose to analyze some of them with respect to several criteria considered as crucial for designing an effective SIR approach.

3.1 Introduction

There is currently a number of interesting research work done to improve the IR process with information coming from social networks. This is commonly known as Social Information Retrieval [16]. We intend here to provide a clear understanding regarding the various state of the art efforts performed in the area of SIR, and in this perspective, our contributions are the following:

1. An objective review of some of the most important research contributions and existing tools in this area to understand the principles of SIR as they are currently formulated.
2. We categorize these contributions based on a taxonomy that we propose in order to structure this wide area.
3. Finally, we propose an analysis of some of these contributions and tools with respect to several criteria considered as crucial for designing an effective SIR approach.

The underlying study is certainly useful for researchers in this area to understand trends, challenges and expectations of a SIR approach. This chapter is organized as follows: In Section 3.2, we introduce our taxonomy of SIR approaches and tools, while describing each of these categories in Sections 3.3, 3.4, and 3.5. In Section 3.6, we present an analysis of some of these contributions and tools based on several

16. SIR has been introduced in Section 2.3.

dimensions that we propose. Finally, we give some future directions in Sections 3.7 and we conclude in Section 3.8.

3.2 A Taxonomy for Social Information Retrieval

There are many contributions in the area of SIR. Each of these contributions considers a particular social network type, and uses social information to perform an IR task differently. For example, on the one hand, tags in folksonomies have been found useful for Web search, personalized search and contextualized enterprise search. Indeed, Gupta et al. [GLYH10] provide a survey summarizing different properties of tags along with its usefulness for IR like tag semantics, recommendations using tags, tag profiling, etc. Also, Heymann et al. [HKGM08] analyze folksonomies, and conclude that social bookmarking systems can provide search results not currently provided by a conventional search engine (Approximately 25% of URLs posted by users are new, unindexed pages). On the other hand, micro-blogging systems like *Twitter* have been also found useful to users to share or submit specific questions to be answered by friends, families, colleagues, or even an unknown person (using a hashtag for a specific topic).

From these two examples, we see that different models of social networks have been used differently for IR tasks. Thus, through the bibliographical study that we have performed, we distinguished several SIR approaches that can be categorized according to the way social information is used. Therefore, we propose a taxonomy for grouping these different initiatives and building a common understanding of this area. Figure 3.1 summarizes this taxonomy of SIR models, which is mainly composed of three categories:

1. *Social Web Search*, in which social information is used in order to improve the classic IR process, e.g. documents re-ranking, query rewriting, user profiling, etc. We discuss this category of SIR approaches in Section 3.3.

2. *Social Recommendation*, in which the user's Social Network is used to make recommendation, e.g. using a social trust network [MYLK08]. This category is discussed in Section 3.4.

3. *Social Search*, in which it is a matter of finding information with the assistance of social resources, such as by asking friends, reference librarians, or unknown persons on-line for assistance [MTP10]. This third category is discussed in Section 3.5.

Several contributions closely related to the area of Social Information Retrieval are discussed in this chapter. The objective is not to discuss the whole set of contributions but to point the closest ones as illustrative contributions. In the following, we discuss and detail these different categories, while giving some illustrative examples.

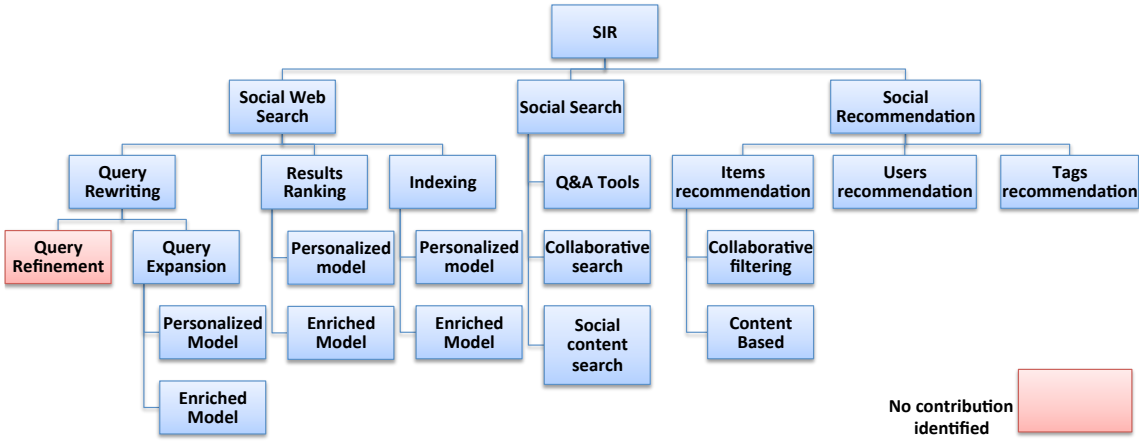


Figure 3.1 – A Taxonomy for Social Information Retrieval Models.

3.3 Social Web Search

We consider this category to include techniques for improving the classical Web search (classic IR process described in Section 2.2.1) using social information. In existing IR systems, queries are usually interpreted and processed using document indexes and/or ontologies, which are hidden for users. The resulting documents are not necessarily relevant from an end-user perspective, in spite of the ranking performed by the Web search engine.

To improve the IR process and reduce the amount of irrelevant documents, there are mainly three possible improvement tracks: (i) query reformulation using extra knowledge, i.e. expansion or refinement of the user query, (ii) post filtering or re-ranking of the retrieved documents (based on the user profile or context), and (iii) improvement of the IR model, i.e. the way documents and queries are represented and matched to quantify their similarities. In Social Web Search, we consider the use of social information in these three tracks.

3.3.1 Query Reformulation

In IR systems, users express generally their needs through a set of keywords that summarize their information needs. Thus, different users are expected to use different keywords to express the same need (e.g. synonyms), and vice versa (i.e. the same keywords can be used by different users to express different information needs). Query reformulation can then bring a solution to this problem. It is defined as follows:

Definition 3.1. [Query reformulation] *Query Reformulation* is the process which consists of transforming an initial query q to another query q'. This transformation may be either a *refinement* or an *expansion*. Query *refinement* reduces the query such that useless information is removed, while query *expansion* adds new information to the initial query to make it less ambiguous.

To the best of our knowledge, there are no contributions yet in query refinement using social information, but all existing works are focusing on social query expansion. In this latter, we distinguish two types of approaches: (i) *enhancing the query expansion terms source*, and (ii) *personalizing the query expansi*on.

3.3.1.1 Enhanced Query Expansion Datasource

In traditional query expansion methods, the database used for the expansion is often constructed according to the comparison between terms' distributions in the retrieved documents and in the whole document collection. In this first category, this database of terms is enhanced using social information without any personalization. Therefore, the underlying idea is to leverage the interactions of users with the system to implicitly and collaboratively build a database of terms, which is expected to feed the expansion process. This will yield a user-based vocabulary source for query expansion. Many methods have been proposed in this area. Biancalana et al. [BMS08] proposed *Nereau*, a query expansion strategy where the co-occurrence matrix of terms in documents is enhanced with meta-data retrieved from social bookmarking services. The system can record and interpret users' behavior, in order to provide personalized search results, according to their interests in such a way that allows the selection of terms that are candidates of the expansion based on original terms inserted by the user.

Lioma et al. [LBM09] provide Social-QE by considering the query expansion (QE) as a logical inference and by considering the addition of tags as an extra deduction to this process. Finally, Lin et al. [LLJY11] propose to enrich the source of terms expansion initially composed with relevant feedback data with social annotations. In particular, they propose a learning term ranking approach based on this source in order to enhance and boost the IR performances.

3.3.1.2 Personalized Query Expansion

In query expansion, providing merely a uniform expansion to all users is often not really suitable nor efficient as relevance of documents is relative for each user [PSC+02]. Thus, a simple and uniform query expansion is not enough to provide satisfactory search results for each user. Personalized social query expansion refers to the process of expanding the same query differently for each user using social information.

Example 3.1. Let's consider the query $q =$ *"Computer science"*, the user Bob may have the expanded query $q' =$ *"Computer science technology programming java"*, whereas the expanded query $q' =$ *"Computer science technology Internet information"* may be more suitable for Alice, depending on their topics of interest.

Several efforts have been made to tackle this problem of personalized query expansion, in particular in the context of folksonomies. Hence, authors in [BCK+08,

SCK+08] consider SIR from both the query expansion and results ranking. In summary, this strategy consists of adding to the query q, k possible expansion tags with the largest similarity to the original tags in order to enrich its results. For each query, the query initiator u, ranks results using BM25 and tag similarity scores. Bertier et al. [BGLK09] propose *TagRank* algorithm, an adaptation of the *PageRank* algorithm, which automatically determines which tags best expand a list of tags in a given query. This is achieved by creating and maintaining a *TagMap* matrix, a central abstraction that captures the personalized relationships between tags, which is constructed by dynamically computing the estimation of a distance between taggers, based on cosine similarity between tags and items.

3.3.2 Results Ranking

In IR, ranking results consists in the definition of a ranking function that allows quantifying the similarities among documents and queries. We distinguish two categories for social results ranking that differ in the way they use social information. The first category uses social information by adding a social relevance to the ranking process, while the second uses it to personalize search results.

3.3.2.1 Ranking Using Social Relevance

Several approaches have been proposed to improve document ranking using social relevance. Social relevance refers to information socially created that characterizes a document from a point of view of interest, i.e. its general interest, its popularity, etc. Two formal models for folksonomies and ranking algorithm called *folkRank and SocialPageRank* have been proposed in [HJSS06b] and [BXW+07] respectively. Both are an extension of the well-known *PageRank* algorithm adapted for the generation of rankings of entities within folksonomies. *SocialPageRank* intends to compute the importance of documents according to the mutual enhancement relation among popular resources, up-to-date users and hot social annotations. In the same spirit, Takahashi et al. [TK08, TK09] propose *S-BIT* and *FS-BIT*, extensions of the well-known HITS [Kle99] approach. Finally, Yanbe et al. [YJNT07] proposed *SBRank*, which indicates how many users bookmarked a page, and use the estimation of *SBRank* as an indicator of Web search. All these algorithms are in the context of folksonomies, and a number of them are reviewed and evaluated in [AHK08].

3.3.2.2 Personalized Ranking

In general, users have different interests, different profiles, and different habits. Consequently, in an IR system, providing the same documents sorted in the same way is not really suitable. Thus, a personalized function to sort documents differently according to each user is expected to improve search results.

Several approaches have been proposed to personalize ranking of search results using social information [BCK+08, CZG+09, NM07, VCJ10, WJ10, XBF+08]. Almost all these approaches are in the context of folksonomies and follow the same idea that the ranking score of a document d retrieved when a user u submits a query q is driven by: (i) a term matching process, which calculates the similarity between q and the textual content of d to generate a user unrelated ranking score; and (ii) an interest matching process, which calculates the similarity between u and d to generate a user related ranking score. Then a merge operation is performed to generate a final ranking score based on the two previous ranking scores. A number of these algorithms are reviewed and evaluated in Chapter 6, while considering different social contexts.

3.3.3 Indexing and Modeling Using Social Information

With the advent of the social Web where all users are contributors, web pages are associated to a social context that can tell a lot about their content (e.g social annotations). Consequently, *social contextual summarization* is required to strengthen the textual content of web pages. Several research work ([ZYW+09, CRYT10, DEFS06, BFNP08]) reported that adding a tag to the content of a document enhances the search quality as they are good summaries for documents [BFNP08] (e.g. document expansion [SP99, HKGM08]). In particular, social information can be useful for documents that contain few terms where a simple indexing strategy is not expected to provide a good retrieval performances (e.g. the *Google homepage* [17]).

Throughout our analysis of the state of the art, we noticed that social information has been used in two ways for modeling and enhancing document representation: (i) either by adding social metadata to the content of documents, e.g. document expansion, or (ii) by personalizing the representation of documents, following the intuition that each user has his own vision of a given document.

3.3.3.1 Enriched Documents

Some approaches investigate the use of social metadata for enriching the content of documents. In [CRYT10, CRYT09, DEFS06, CZ09], authors index a document with both its textual content and its associated tags modeled as in the Vector Space Model (VSM). However, each approach uses a different algorithm for weighing social metadata, e.g. tf-idf [CRYT10, CZ09], term quality [CRYT10, CRYT09], etc. Also, Zhang et al. [ZYW+09] propose a framework to enhance document representation using social annotations. The framework consists in representing a Web document in a dual-vector representation: (i) enhanced textual content vector and (ii) enhanced social content vector. Each component is calculated from the other.

17. http://www.google.com/
There are only a very few terms on the page itself but a thousands of annotations available on *delicious* are associated to it. Eventually, social annotations of the *Google* homepage are more useful for indexing.

3.3.3.2 Personalized Documents Representation

Given a document, each user has his own understanding of its content. Therefore, each user employs a different vocabulary and words to describe, comment, and annotate this document (see Figure 5.1). For example, if we look at the homepage of Youtube, a given user can tag it using "video", "Web" and "music" while another can tags it using "news", "movie", and "media". Hence, Amer-Yahia et al. [AYBLS08] investigate efficient top-k processing in collaborative tagging sites. The idea is that the score of an answer is computed as its popularity among members of a seeker's network. Basically, the solution is to create personalized indexes based on clustering strategies, which achieve different compromises between storage space and processing time.

In summary, in this section we discussed the Social Web Search category of SIR. As it has been described above, social Web search refers to the use of social information to improve the classic IR process (presented in Section 2.2.1). We showed that social information has been used at different levels in the IR process: either to globally enrich the IR process or to personalize it. In the next section, we present Social Recommendation in which recommendation is done based on the user's social network.

3.4 Social Recommendation

The second category of SIR models considers filtering and recommendation fields (e.g. content-based filtering, collaborative filtering, recommender systems). Basically, recommendation aims at predicting the interest that users would give to an entity they had not yet considered explicitly. There are two main methods of recommendation: (i) an approach based on recommending items that are similar to those in which the user has shown interest in the past, which is known as *"content-based"* approach (CB), and (ii) an approach that intends to recommend items to the user based on other individuals who are found to have similar preferences or tastes, which is known as *"collaborative filtering"* approach (CF). We define social recommendation as follows:

Definition 3.2. [Social Recommendation] Social recommendation is a set of techniques that attempt to suggest: (i) items (e.g. movies, music, books, news, web pages), (ii) social entities (e.g. people, events, groups), or (iii) topics of interest (e.g. sport, culture, cooking, etc.) that are likely to be of interest to the user through the use of social information.

On the one hand, in recent years, many personalized recommendation features based on the user's social network have been developed and integrated to popular web sites. This is mainly done by prompting the user to connect his social networks' accounts to their services; the ultimate goal being collecting as much data as possible that characterize the user. For example, videos recommendation in *YouTube* based

on the *Google+* profile, or movies recommendation on *IMDb* through the registration using a *Facebook* account. This functionality has been reported to lead to an improvement of the recommendation services.

On the other hand, many social platforms understood the power of learning over their data for building recommender services. For example, targeted advertising in some social platforms like *Facebook*, group recommendation again on *Facebook,* follower recommendation on *Twitter*, or topics and web pages recommendation on *delicious*. Moreover, there are other social web services, whose recommendation is at the heart. For example, social news aggregation services, like *Digg*, which presents stories expected to be most interesting to a user, based on preferences of similar users. It is clear that all these services have been improved by exploiting their social information.

In the following, we categorize social recommender systems according to the type of output they intend to recommend: (i) items recommendation, (ii) users recommendation, and (iii) topics recommendation.

3.4.1 Items Recommendation

Items recommendation has probably attracted the most attention in recommender systems. Classical items recommendation methods are based on the assumption that users are independent entities and identically distributed. Thus, they don't suppose any additional structure, including the social network that surrounds users. This doesn't reflect the real behaviors of users, since they normally ask friends for recommendations before acting, e.g. buying a product. Many researchers have then started exploring social relations to improve recommender systems. In collaborative filtering based approaches, Liu and Lee [LL10] proposed very simple heuristics to increase recommendation effectiveness by combining social networks information. Guy et al. [GZC+09] proposed a ranking function of items based on social relationships. This ranking function has been further improved in [GZR+10] to include social content such as related terms to the user.

Another common approach to CF attempts to factorize an (incomplete) matrix R of dimension $I \times J$ containing observed ratings $r_{i,j}$ (which represents the rate of the user i to the item j) into a product $R \approx U^T V$ of latent feature matrices U and V. In this context, following the intuition that a person's social network will affect his behaviors on the Web, Ma et al. [MYLK08] propose to factorize both the users' social network and rating records matrices. The main idea is to fuse the user-item matrix with the users' social trust networks by sharing a common latent low-dimensional user feature matrix. This approach has been improved in [MKL09] by taking into account only trusted friends for recommendation while sharing the user latent dimensional matrix. Almost a similar approach has been proposed in [JE10] and [YSL12] who include in the factorization process, trust propagation and trust propagation with inferred circles of friends in social networks respectively. In this same context, other approaches have proposed to consider *social regularization terms* while factorizing the rating matrix.

The idea is to handle friends with dissimilar tastes differently in order to represent the taste diversity of each user's friends [MZL+11, NST+12, YPL11].

3.4.2 Users Recommendation

As said above, social network platforms have adopted the strategy of suggesting friends (or group of friends) to increase the connectivity among their users. Hence, content-based approaches were proposed in [CGD+09] to match the content of user profiles and determine user similarities for recommendation. Groh et al. [GE07] generated user neighborhood information from known social network structures and demonstrated that collaborative filtering based on such neighborhood outperforms classic collaborative filtering methods. Symeonidid et al. [SNM10] proposed a ternary semantic analysis unified framework to perform users recommendation. Guy et al. [CGD+09] describe an user interface for providing users with recommendations of people to invite into their social network. The approach is based on aggregated information collected from various sources. Hannon et al. [HBS10] utilize content and collaborative-based approaches to evaluate a range of different user profiling and recommendation strategies. Many of these strategies have been implemented in [HMS11]. Finally, in the context of tagging systems, Wang et al. [WLF11] propose to connect users with similar tastes by measuring their similarities based on the tags they share in an inferred network of tags.

3.4.3 Topics Recommendation

Recently, topic and tag recommendation has attracted significant attention to produce hotlists of high quality to users. Recommendation of tags also allows users to choose the right tags as tagging is not constrained by a controlled vocabulary and annotation guidelines. Hence, the work in [JMH+08] provides a comprehensive evaluation and comparison of several state of the art tag recommendation algorithms in three different real world datasets. A content-based collaborative filtering technique has been proposed in [Mis06] to automate tags assignments to blogs. Hotho et al. [HJSS06b] propose to project the three-dimensional correlations to three 2D correlations. Then, the two-dimensional correlations are used to build conceptual structures similar to hyperlink structures that are used by Web search engines. The work in [HJSS06a, SNM10] have shown to generate high quality tags recommendations that outperform baseline methods such as the most-popular models and collaborative filtering [JMH+08]. Also, Krestel et al.[KFN09] proposed an approach to use Latent Dirichlet Allocation to expand tag sets of objects annotated by only a few users.

Finally, several approaches have been proposed making these three types of recommendations, i.e. social items, users, and topics recommendations, unified under one framework. Carmel et al. [CRYT10] propose a framework for social bookmarks weighting, which allows estimating the effectiveness of each of the bookmarks individually for several IR tasks. To do this, they propose several recommendation

strategies such as tag recommendation, user recommendation, and document recommendation. The obtained values from the three strategies are merged in order to effectively estimate the bookmarks quality. In the same spirit, the framework proposed by Symeonidid et al. [SNM10] acts in the same way. This framework models the three types of entities by a 3-order tensor, on which multiway latent semantic analysis and dimensionality reduction is performed using both the Singular Value Decomposition method and the Kernel-SVD smoothing technique. Also, Wei et al. [WHL11] propose to leverage a quaternary relationship among users, items, tags and ratings to provide recommendation. They propose a unified framework for user recommendation, item recommendation, tag recommendation and item rating prediction by modeling the quaternary relationship among users, resources, tags, and ratings as a 4-order tensor and cast the recommendation problem as a multi-way latent semantic analysis problem.

In summary, in this section we discussed the social recommendation category of SIR. We showed that with the emergence of the social Web, the recommendation domain has evolved to handle and recommend many type of entities. It is clear that social networks represent a valuable source of information to improve and develop effective and efficient recommendation algorithms. In the next section, we present social search, a new paradigm to perform IR tasks.

3.5 Social Search

Social platforms like *Twitter* and *Facebook* allow users to share and publish information with their friends and often with general public. In addition to this, users use them to answer very precise and highly contextualized queries, or queries for which the relevant content has not been authored yet, e.g. asking about a conference event using its hashtag on *Twitter*. We refer to such a process of finding information as *Social search*, and we define it as follows.

Definition 3.3. [Social search] Social search is the process of finding information only with the assistance of social entities, by considering the interactions or contributions of users.

Thus, social search is associated to platforms that are defined as search engines specifically dedicated to social data management such as Facebook. The main ingredient to perform a social search is the user interactions, including: (i) social content (e.g. comments, tweets, etc.), and (ii) social relations (e.g. finding a person with a certain expertise). Hence, social search systems index either social content and offer a means for users to search that content [TRM11], or social relations and allow the user finding persons, who are likely to respond to specific needs [HK10]. We divide social search into three main categories detailed in the following.

3.5.1 Social Content Search

Social platforms allow users to provide, publish and spread information, e.g. commenting or tweeting about an event. In such a context, a huge quantity of information is created in social media, which represents a valuable source of relevant information. Hence, many users use social media to gather recent information about a particular content by searching collection of posts and status. Therefore, social content search systems come as a mean to index content explicitly created by users on social media and provide a real-time search support [JCC10].

There are several social content search engines, which index real-time content spreading systems. This includes *TwitterSearch* [18], *Social Bing* [19], *collecta* [JLW+11], *OneRiot* [One09] [20], etc. Social content search systems deal with a different kind of content than classic search engines. Indeed, posts and statuses published on social media are often short, frequent, and do not change after being published, while web pages are rich, generated more slowly, and evolve after creation [TRM11]. Dealing with such content is challenging, because it requires real-time and recency sensitive queries processing. Sensitive query refers to a query where the user expects documents, which are both topically relevant as well as fresh [DZK+10]. A study has been performed by Teevan et al. [TRM11] that give an overview of "What is the motivation behind a user to use a social content search system rather than a classic search engine?". This study reveals that social content search systems are interrogated with queries, which are shorter, more popular, and less likely to evolve as part of a session than Web queries. The main goal is to find temporally relevant information (e.g. breaking news, real-time content, and popular trends) and information related to people (e.g. content directed at the searcher, information about people of interest, and general sentiment and opinion).

3.5.2 Social Question/Answering (Q&A)

Despite the development of techniques and methods for Web search assistance such as navigational queries [Bro02] and query auto-completion, for helping users to express their needs, many queries still remain unanswered. Dror et al. [DKMS11] argue that this is mainly due to two reasons: (i) the intent behind the query not being well expressed/captured, and (ii) the absence of relevant content.

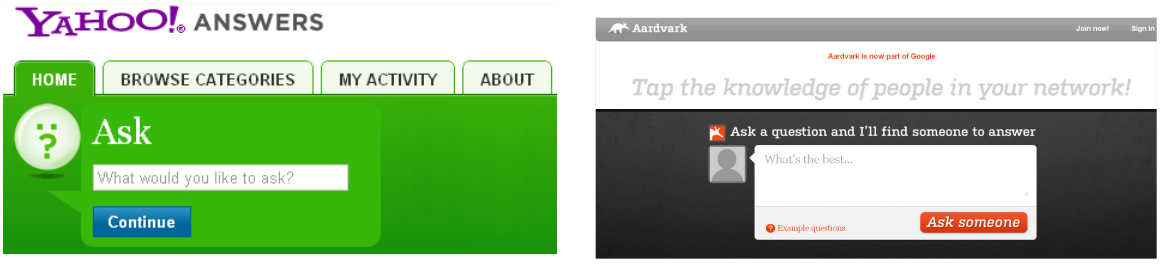
To tackle these issues, *Question/Answering Systems (Q&A)* have emerged to connect people for helping each other answering questions. Examples of such systems include *Yahoo Answer!* [21], *WikiAnswers* [22], and *Aardvark* [23]. Basically, Q&A systems provide a

18. http://search.twitter.com/
19. http://www.bing.com/social
20. OneRiot has been acquired by Walmart in September 2011. This service is no more available.
21. http://answers.yahoo.com/
22. http://wiki.answers.com/
23. Aardvark has been acquired by Google on February 11, 2010. In September 2011, Google announced it would discontinue a number of its products, including Aardvark.

means for answering several types of questions such as recommendation, e.g. building a new play list, any ideas for good running songs?, opinion seeking, e.g. *I am wandering if I should buy the Kitchen-Aid ice cream maker?*, factual knowledge, e.g. *Does anyone know a way to put Excel charts into LaTeX?*, problem solving, e.g. *How do I solve this Poisson distribution problem?*, etc. Morris et al. [MTP10] conducted a survey in which they study the type of asked questions, the frequency of these questions, and the motivations for users asking their social network rather than using a traditional search engine. Figure 3.2 shows the web UI for users to ask their questions on *Yahoo Answer!* and *Aardvark.*

Aardvark was certainly one of the most promising Q&A systems [HK10]. It allowed connecting users with friends or friends-of-friends who are able to answer their questions. Users submitted questions via the Aardvark Website, email or instant messenger (see Figure 3.2b). Aardvark was able to identify and facilitate a live chat or email conversation with one or more topic experts in the asker's extended social network. Users was also able to review questions and answers history and other settings on the Aardvark website. The challenge in such systems lies in finding the right person to satisfy a user's information need, in contrast to traditional search engines where the challenge lies in finding the documents that are likely to satisfy user's query.

The main problem facing such systems is in the response time as well as the quality of the answers. For example, Zhang et al.[ZAAN07] report that on the Java Developer Forum, the average waiting time of a high expertise user to get a reply for a question is about 9 hours, compared with 40 minutes for a low expertise user. As for Microsoft's Live Q&A site, Hsieh and Count [HC09] state that 80% of queries receive an answer, with an average of response time of 2 hours and 52 minutes. As for Aadvark [HK10], 87,7% of questions received at least 1 answer, and 57,2% received their first answer in less than 10 minutes. Dror et al. [DKMS11], study a dataset of 4 month period of Yahoo Answer! interactions, and state that the average time for answering queries is 10 minutes, while almost all answers were given within 60 minutes of the question's creation time.



(a) Yahoo! Answers

(b) Aardvark.

Figure 3.2 – Examples of two social search engine that allow users to both submit questions to be answered and answer questions asked by other users.

3.5.3 Social Collaborative Search

One of the weaknesses of search engines available today (e.g. Google, Yahoo!, Bing) is the fact that they are designed for a single user who searches alone. Thus, users cannot benefit from the experience of each other for a given search task. Morris [Mor07, Mor08] conducts a survey on 204 knowledge workers in a large technology company in which she revealed that 97% of respondents reported engaging in one of collaborative search task described in the survey. For example, 87,7% of respondents reported having watched over someone's shoulder for query suggestion, and 86,3% of respondents reported having e-mailed someone to share the results of a Web search.

In such a context, Morris [MH07] developed *SearchTogether*, a collaborative search interface, where several users who share an information need collaborate and work together with others to fulfill that need. The authors discuss the way *SearchTogether* facilitates collaboration by satisfying criteria like awareness, division of labor, and persistence. Similarly, Filho et al. [FOdG10] proposed *Kolline*, a search interface that aims at facilitating information seeking for inexperienced users by allowing more experienced users to collaborate together.

Finally, Paul and Morris [PM09] investigate *sensemaking* for collaborative Web search, which is defined as the act of understanding information. The study revealed several themes regarding the sensemaking challenges of collaborative Web search, e.g. Awareness, Timeliness and sensemaking hand-off. Based on their finding, they proposed *CoSense*, a system that supports *sensemaking* for collaborative Web search tasks that provides enhanced group awareness by including a time-line view of all queries executed during the search process. Even though these features help to enhance participants' communication and *sensemaking* during their search activities, users still have to sort among different documents and analyze them one by one to find relevant information.

In summary, in this section we discussed the Social Search category of SIR. We showed that the search paradigm has evolved to provide user's tools and methods for asking more sophisticated and contextualized queries. Social content search systems allow to mine and find relevant information within posts and users comments, while Q&A systems allow users to answer very contextualized queries, whereas social collaborative search systems come as a mean to help users share their experiences and findings. In the next section, we give an overview of many platforms, methods and approaches of SIR as well as their potential drawbacks. The main objective is to understand the impact of the social dimension on the IR process, the weaknesses and the possible future contributions in this area.

3.6 Analysis of SIR approaches & platforms

The objective in this section is to analyze the richnesses and weaknesses of some SIR tools and approaches. Among the discussed SIR tools and methods in the previ-

ous sections, only few of them have spawned to a concrete commercial prototype. The selected tools for analysis are provided in Table 3.1, and their description is provided in the previous sections. We have selected these approaches and tools for three main reasons: (i) they are the most popular ones when this analysis is performed (judging from our research), (ii) they are the most referenced ones and are published in top venue conferences [24], and finally, (iii) this limit is motivated by the fact that our objective is not to analyze all the existing approaches and tools, but the most representative ones as illustration of some underlying principles and ideas.

For this analysis, we choose eight dimensions covering various aspects of the social information used and the problem to be solved. We consider these dimension as crucial for designing an effective and efficient SIR approach. For illustration purposes, we only discuss some of the above tools for each dimension. Table 3.1 summarizes the analysis dimensions and the consideration of the different tools according to these dimensions. In the following sections, we discuss these dimensions, their meaning, their importance, and the degree to which the tools consider them.

3.6.1 Social Networks

This dimension is related to the kind of social networks leveraged by a SIR approach. Each SIR approach relies on almost one kind of social network either by using its social content or by exploiting its social relations. Depending on the considered approach, its application and its purpose, it can use:

— Social networks with symmetric relationships in which users explicitly declare their social relations of friends. Users can also express their opinions, comment news, and share resources. This represents a valuable source of information for user modeling and profiling [HK10, JLW+11], which can be reused to build many interesting services, e.g. services of recommendation, personalization, etc.

— Social networks with asymmetric relationships, which can be divided into two categories: (i) content subscription social networks, e.g. microblogging systems, in which most of the considered SIR approaches relies on using the temporal aspect of these networks, e.g. answering recency sensitive queries [DZK+10], and a (ii) trust-based Social Network, e.g. *epinion* [25], in which SIR approaches use the trust degree between users, e.g. for collaborative filtering [MYLK08].

— Social networks with ternary relations, e.g. bookmarking systems. As discussed in Chapter 2, the generated structures in these systems have been proven to be a valuable knowledge for building many SIR approaches. Exploiting annotations and social metadata explicitly provided by users has been used in different way, for example to extract correlated terms [LLJY11], build users' profiles [CZG+09], compute social relevance [BXW+07], enhance documents [ZYW+09], etc.

24. A highly selective conference like SIGIR, WWW, CIKM, VLDB, WSDM, KDD, etc.
25. http://www.epinions.com/

Table 3.1 – Summary of the social dimension in many social web search approaches. (×) means the dimension (i.e. functionality) is provided, (-) means the dimension is not provided. These marks do not imply any "positive" or "negative" information about the tools except the presence or the absence of the considered dimension.

			Social Information Retrieval													
			Social Web Search						Social Recommendation					Social Search		
			QE	Ranking			Indexing		Item		User	Topic		Q&A	Content	Collaborative
			[LLJY11]	[CZG+09]	[BXW+07]	[XBF+08]	[ZYW+09]	[CRYT10]	[MYLK08]	[NST+12]	[CGD+09]	[KFN09]	[SNM10]	[HK10]	[JLW+11]	[MH07]
Social Networks[a]	SRSN		-	-	-	-	-	-	-	×	×	-	-	×	×	-
	ARSN[b]	CS	-	-	-	-	×	-	-	-	-	-	-	×	×	-
		TR	-	-	-	-	-	-	×	×	-	-	-	×	-	-
	TRSN		×	×	×	×	×	×	-	-	-	×	×	-	-	-
	Their own SN		-	-	-	-	-	-	-	-	-	-	-	×	-	×
Social Network Data	Content		×	×	×	×	×	×	-	-	×	×	×	×	×	-
	Structure		-	×	×	-	-	×	×	×	×	×	×	-	-	×
Datasource[c]	DCSN		×	×	×	×	×	×	×	×	×	×	×	×	×	-
	DCW		-	-	-	×	×	×	-	-	-	-	-	-	-	-
	DEPU		-	-	-	-	-	-	-	-	×	-	-	×	-	×
Personalization[d]	Profile		-	×	-	×	-	-	-	-	-	-	-	×	-	-
	CB		-	-	-	-	-	-	-	-	×	-	×	-	-	×
	CF		-	-	-	-	-	-	×	×	×	-	×	-	-	-
Complexity & Adaptability	Scalability		×	×	-	×	×	-	×	×	×	×	-	×	×	×
	Dynamicity		-	×	-	×	-	-	-	-	×	-	-	×	×	×
	Data Sparsity		×	-	×	×	×	×	×	×	-	-	×	×	×	-
	Cold Start		×	×	×	×	×	×	×	×	-	-	-	×	×	-
Socialization			-	-	-	-	-	-	×	×	×	-	×	×	×	×
Privacy			×	×	×	×	×	×	-	-	-	-	-	-	-	-
Industrialization			-	×	-	-	-	-	-	-	-	-	-	×	×	×

[a] SRSN = Symmetric Relationship Social Network; ARSN = Asymmetric Relationship Social Network; TRSN = Ternary Relationship Social Network

[b] CS = Content Subscription; TR = Trust relations

[c] *DCSN* = Data Crawled from Social Networks; *DCW* = Data Crawled from the Web, e.g. Content of Web pages; *DEPU* = Data explicitly provided by users provided by users

— Their own social networks built upon information gathered from: (i) the user, by explicitly providing information about him, (ii) the aggregation and crawling of both social networks and the Web, e.g. building a FOAF (Friend Of A Friend) ontology, which is a machine-readable structure that describes people, or (iii) by an inference process based on the user behavior, e.g. Aardvark [HK10].

3.6.2 Social Data

This dimension concerns the kind of social information leveraged by SIR approaches. We distinguish the following two kinds of social information, which are embedded within a social network:

— Social data content, which is the content generated by users through their interactions and activities, i.e. activities of publishing, annotating, commenting or rating content and entities. This social content is most of the time useful to: enrich entities and thus enhance their logical representation [ZYW+09, CRYT10, DZK+10], extract correlated terms, e.g. through co-occurrence of terms [LLJY11], pull out user profiles [CZG+09, HK10], etc.

— Social relations explicitly or implicitly declared by users. Indeed, social networks exhibit various relationships, between entities of their social graphs. These relationships can be within entities of the same type, e.g. friendship relations between users, similarity between resources, etc., or between entities of different types, e.g. an authorship relation between a user and a document, a relation of description between a term and a user, etc. Social relations are useful for building a SIR approach, and have been used in different ways, e.g. for recommendation [MYLK08, NST+12] by handling trust relations, for extracting correlated terms by leveraging their relations over document [LLJY11], etc.

3.6.3 Data sources

The data sources dimension refers to the source of information used by SIR approaches. These latter are most of the time not only based on social data sources, but also on other sources of information like:

— The content of web pages, which contains valuable information that can be extracted by performing classic textual treatments [LLJY11, CRYT10, ZYW+09].

— Ontologies, e.g. FOAF ontology (Friend Of A Friend), which can be used to infer a social network [Mik07a].

— Information explicitly provided by users. Some SIR approaches ask users to explicitly provide information to enrich their internal data model [HK10].

Note that at the basis, social information is only used to improve and enhance information retrieval tasks. Hence, we believe that the use of trusted sources of information (e.g. information provided by authors of web pages) is necessary to the proper

functioning of SIR approaches (actually, almost all SIR approaches combine such data sources [CRYT10, ZYW+09]). A trade-off should be found to properly weight each data source, while avoiding overfitting.

3.6.4 Personalization

As discussed in the previous sections, some SIR approaches are based on personalization, e.g. [CZG+09, HK10, BGLK09, XBF+08]. Personalization allows differentiating between individuals by emphasizing on their specific domains of interest and their preferences. It is a key point in IR and its demand is constantly increasing by numerous users for adapting their results [Bel08]. Several techniques exist to provide personalized services among which: (i) the user profiling, (iii) content based approaches, and (ii) collaborative based approaches.

3.6.4.1 Profile Based Approaches

A user profile is a collection of data describing a specific user, which are explicitly or implicitly provided by him. Therefore, a profile refers to the digital representation of a person's identity, which includes mainly the description of his characteristics and his domain of interest and expertise. We distinguish two types of profiles that differ in the way they are constructed:

Profile Constructed Offline Some SIR approaches construct profiles offline and maintain them incrementally, which make them more efficient regarding the execution time [HK10]. However, profiles computed offline decreases the dynamicity of the approach since new data are not instantly taken into account.

Profile Constructed on the Fly (online) In contrast, profiles computed online increase the dynamicity of an SIR approach and its efficiency, while degrading the execution time [CZG+09, XBF+08].

3.6.4.2 Content-Based Approaches

Personalization in content based approaches aims to provide to users, information similar to that they previously consumed. Personalization using a content based approach is used in recommender systems [GZR+10, SNM10].

3.6.4.3 Collaborative-Based Approaches

Personalization based on collaborative filtering process aims to provide to users, information consumed by many similar users [MYLK08, MKL09, MZL+11]. This process of collaborative filtering can be summarized to two steps:

1. Look for users who share the same interests and behavior with a given user, i.e. the user who is currently using the SIR approach.
2. Provide to the considered user, contents and information based on the users found in step 1.

3.6.5 Complexity and Applicability

The complexity of an approach tends to characterize it from several perspectives of its applicability. We propose several dimensions to describe and characterize the complexity and the applicability of an approach.

3.6.5.1 Scalability

Scalability is the ability of an approach to continue to function well when its context is changed. This refers to its ability to scale to a very large dataset (number of users, resources, objects, documents, etc.) while continuing to meet the needs of users (both relevance and precision) and to take full advantage of it in terms of performance, e.g. execution time and relevance of information. Some SIR approaches are able to scale to a very large dataset [MYLK08, MZL+11, NST+12], while other not [CRYT10] because of their algorithms' complexity.

3.6.5.2 Dynamicity

Dynamicity refers to the ability of approaches to consider new data and to quickly update their model. Considering new data is a key problem for SIR approach since they are based on social information, which is growing quickly with the intense activity of users (users are constantly in the process of commenting, editing, publishing and sharing information). Some SIR approaches have a model, which is rather flexible to update [CZG+09, HK10], whereas other approaches not, e.g. approaches based on machine learning techniques, which are among the most difficult to update since we have to rebuild each time the model in order to consider new data [MYLK08, MKL09, MZL+11, LLJY11].

3.6.5.3 Data Sparsity

Data sparsity is a term used to designate how much data we have for the dimensions of a dataset. The term of sparse data is most of the time associated with matrices, where a sparse matrix is a matrix populated primarily with zeros [SB02]. Therefore, this dimension refers to the ability of an approach to process over sparse data. Indeed, sparse data leads to the problem of poor results quality since there is no enough information to process. However, some approach handle efficiency this problem by considering other sources of data, e.g. content of web pages or other social networks [HK10, LLJY11, MYLK08].

3.6.5.4 Cold Start

The cold start is a potential problem of systems to handle effectively new entities, e.g. users, items, etc. In other words, it concerns the issue that the system cannot draw any inference for users or items about which it has not yet gathered sufficient information. Recommender systems are among the most affected systems by the cold start problem since they need enough information to make predictions. However, many approaches are dealing efficiently with this problem by relying on other data sources [CZG+09, MYLK08, MKL09, MZL+11].

3.6.6 Socialization

Some SIR approaches put users in contact, and encourage them to socialize and collaborate in order to satisfy a particular goal of information needs. This dimension refers to this particularity of socialization between users.

Thus, many approaches put users in contact in order to benefit from the experience of each other. Especially, many social recommender systems offer this feature, since they provide explanations of why a recommendation has been done, and what yielded the recommendation [GZC+09, CGD+09, GZR+10]. Then, if the recommendation is based on the experience of other users, e.g. rating of other users, the current user can contact them to obtain other information and precisions. Also, as stated before, social search approaches are exclusively based on the assistance of social entities. This will automatically put users in contact as it is the case in *Q&A* systems (Aadvark [HK10]), and social collaborative search systems (*S*earchTogether [MH07], Kolline [FOdG10], and sensemaking [PM09]).

3.6.7 Privacy Management

Social information is sensitive to the privacy of users. Some approaches don't consider the privacy of users, as they spread sensitive information about users. Even if users make their social accounts public, we believe that reusing these data in other value-added services is still a user privacy problem.

Hence, probably social recommender systems and social search systems are the most related and exposed to users' privacy problems. Examples where we need to divulge information about users include:

(i) In recommender systems, we need to justify why a recommendation has been done.

(ii) In *Q&A* system, we need to justify why we put two users in contact.

(iii) In social content search systems, we show who said what on what.

(iv) In social collaborative search systems, we show who searched what.

Finally, many approaches are using social information in back-office, and thus, don't spread sensitive information about users. This includes mainly social Web search approaches [LLJY11, CZG+09, CRYT10, ZYW+09, XBF+08].

3.6.8 Industrialization

Developing effective and efficient algorithms is good, but putting them in action is better. Hence, this final dimension is related to an aspect of production and commissioning. Many SIR approaches studied in this chapter have spanned to a concrete and commercial prototype, e.g. Aadvark [HK10], *S*earchTogether [MH07], or collecta [JLW+11], while other approaches are still just a research contributions.

3.7 Discussion and Future Directions

In this section, we make a general discussion on SIR with respect to the taxonomy proposed in Section 3.2. Our main goal is to give possible improvement tracks, which we can consider for future works.

In the social Web search track, especially in the social query rewriting part, we report that there is no approach that considers query reduction. Query reduction is a technique to reduce long queries, to queries, which are shorter and more effective. Hence, investigating a social query rewriting approach by considering both query expansion and reduction, seems to be promising and provide good research perspectives. This can be by providing users with a way to suggest them other queries that can better match their requirements. For example, the query "free movies" can be rewritten as "streaming action movies" by replacing the term "free" with "streaming" and expanding the query with the term "action" following the intuition that this user is most likely to appreciate action movies.

Regarding the IR modeling part (documents representation and ranking functions), we believe that the temporal dimension is a key aspect, which hasn't been deeply investigated. This includes considering the evolution of users' behavior, and profiles in time. Indeed, users are expected to evolve in time, which make their taste different. We believe that IR systems should constantly learn information about users in order to adapt their results.

With regard to the social recommendation track, this topic has been deeply investigated, whether for items, tags or users recommendation. Items recommendation is probably the part that attracted the most attention. However, we also believe that the temporal dimension of recommender systems can help to tackle the problems of: (i) hot topics, i.e. news, fresh information, (ii) the evolution of user profiles along time, i.e. the user interests evolve and change along time, and (iii) the diversity of information, i.e. in order not to annoy users with similar information. Indeed, for the first problem, information is time-dependent, meaning that it attracts much attention at a given moment and will be quickly forgotten after a while. Many users of social

media state that the freshness of information is a key point in a recommender system. The second problem deals with the evolution and the update of user profiles. For example, the opinion of a user concerning something may change in time when he grows, or reads news about this thing. The third problem is also a feeling that users have when they use social media (Facebook in particular). Most of the time, when a recent information appears, all users begin to publish articles that deal with the same information, and users are quickly overwhelmed by similar information published mutually by each other. We believe that, at a given time, the recommender system should know that a given user is already aware about this information and consequently it should be hidden [NST+12].

Finally, with the advent of social networks, many tools and approaches have emerged to deal with the weakness of classic search engines. This includes social content search engines (for opinion seeking), query recency search (Q&A tools for contextualized search), and collaborative search engine (for collectively answering a common information need). In these areas, many improvement still possible like finding the right persons for answering a given question in Q&A tools or how to find users who share common information needs and relate them to work together to fulfill this need.

3.8 Conclusion

In this chapter, we proposed a taxonomy to classify and categorize SIR approaches into three main categories, namely: social Web search, social recommendation, and social search. We showed that these three sub-categories are fundamentally different in the way they leverage and use social information. Many approaches have been proposed to improve the classic IR process, while others proposed new search paradigms based on the socialization between users.

Social platforms represent valuable sources of information and knowledge that can be reused to improve many services (especially search services). On the one hand, initiatives like OpenID, and the Mashup concept are promoting the possibility to social platforms to share their data with other applications, e.g. the full user name, the profile's picture, the gender, the username and user id (account number), and even the list of friends and contacts. On the other hand, users are encouraged to share and publish content on social platforms, and make their social information publicly accessible. Therefore, this social dimension of the Web began to attract the attention of many researchers in order to contribute to the IR topic.

We discussed many SIR approaches with respect to some dimensions that we judge to be essential in order to assess the robustness and the effectiveness of a SIR approach. We consider these dimensions, as factors that indicate to which extent a SIR approach can be applicable, and in which context, i.e. size of the data, sparsity of the data, etc. Finally, we discussed some interesting improvement for some SIR categories for a research perspective. Currently, many of the proposed improvement tracks can

be possible while other are not (mainly for complexity reasons).

Finally, in this thesis, we focus on social Web search. We believe that the new generation of IR systems should fully leverage the social dimension of the Web, while providing personalized search services. Indeed, as said above, personalization is a key point in IR and its demand is constantly increasing by numerous users for adapting their results [Bel08]. Hence, In this PhD thesis, we address the social personalized search in its entirety. This includes the three possible improvement tracks, which we described previously: (i) query reformulation using extra knowledge (Chapter 4), (ii) post filtering or re-ranking of the retrieved documents (Chapter 6), and (iii) improvement of the IR model (Chapter 5). All the proposed algorithms in this PhD thesis are fully driven by the dimensions described in Section 3.6. The main idea is to find a trade off to satisfy these criteria while designing our algorithms, in order to make them as effective and efficient as possible.

Chapter 4

Personalized Social Query Expansion Using Folksonomy

This chapter introduces our proposal for query expansion using social information. While considering social tagging systems, we propose an approach that considers: (i) the semantic similarity between tags composing a query, (ii) a social proximity between the query and the user for a personalized expansion, and (iii) a strategy for expanding, on the fly, user queries.

4.1 Introduction

There exists different way to improve the performance of an IR system, among which, a pre-processing of queries such as query expansion, which is the focus of this chapter. Hence, query expansion consists of enriching the user's initial query with additional information so that the IR system may propose suitable results that better satisfy the user's needs [Eft96]. This information can be of different natures and sources. Examples are numerous:

— *Contextual information:* Many queries need contextual information like time and location. For example, a restaurant in New York proposed by a search engine when a user in Paris, submits the query *"seafood restaurant"* is completely irrelevant. Especially if the restaurant is closed at the query time. Hence, in many cases, contextual information should be integrated to the query before being processed by a search engine in order to propose suitable results.

— *Ontological knowledge:* Many queries require to be enriched with additional terms in order to propose satisfactory results. For example, synonyms of the query terms should be added in order to retrieve relevant documents that may use different words than those of the query. As an illustration, the query "*data mining*" should also include the terms "*machine learning*" to retrieve and highlight other relevant documents since these two terms are commonly confused.

— *Social knowledge:* In many cases, a query should be enhanced with social information. For example, suppose that a user with a Facebook account is looking for an attraction in Los Angles by submitting the query *"attraction in los angeles"* to a search engine. The system should enhance the query with the opinions of his friends, who have already rated attractions in Los Angeles to determine which kind of attractions may correspond to him. This kind of query expansion may really propose relevant documents to the user by ensuring that the search engine acts as a recommender system.

All the above examples of data sources for expanding queries can be of valuable knowledge. However, in this Chapter, we focus exclusively on the possibility of using social knowledge as a source of implicit feedback information for query expansion. In particular, we propose to use the social data of folksonomies by exploiting both the structure and the content of these valuable data sources. We propose an approach which reuses the users vocabulary (the terms used to annotate web pages) in order to expand user queries in a personalized way and thus, increase their satisfaction regarding the quality of search.

Example 4.1. Suppose that a user is looking for *Java* on *Google*. Depending on the interests of this user, the approach we are proposing may expand this query differently. If this user is more interested in travel, tourism and escapades, we could expand this query to become "*java island*". If this user is more interested in computer science, we could expand it differently. If his interests raise more in Java tools, the query may become "*java open source tools*". Otherwise, if his interest raise more in java language, the query may become "*java programming language*".

The above example gives a good illustration of what personalized query expansion means. Our approach in this work consists in three main steps: (i) determining similar and related tags to a given query term through their co-occurrence over resources and users, (ii) constructing a profile of the query issuer based on his tagging activities, which is maintained and used to compute expansions, and finally, (iii) expanding the query terms, where each term is enriched with the most interesting tags based on their similarities and their interest to the user.

The main contributions of this chapter can be summarized as follows:

1. We propose an approach in which we use social knowledge as explicit feedback information for the expansion process. Reusing such a social knowledge aims at expanding user queries with their own vocabularies instead of using a public thesaurus, which is made by people who are not aware of the individual users needs and expectations.
2. We propose a Personalized Social Query Expansion framework called PSQE. This latter provides a user-dependent query expansion based on social knowledge, i.e. for the same query of two different users, PSQE will provide two different expanded queries, which will be processed by a search engine.

3. Using an evaluation on real data gathered from three different large bookmarking systems, we demonstrate the effectiveness of our framework for socially driven query expansion compared to many state of the art approaches.

Figure 4.1 shows an overview of the approach we are proposing. Note that our contributions reside mainly in how we may leverage the social knowledge of a folksonomy to extract relevant information, and how we use this information for query expansion purpose.

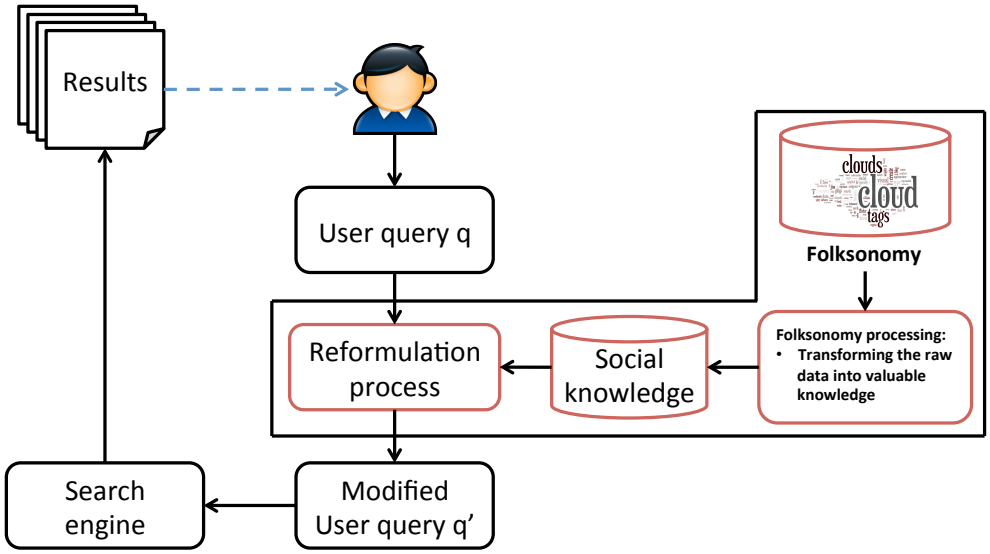


Figure 4.1 – Approach overview

This chapter is organized as follows: in Section 4.2, we give a formal definition of the problem we are tackling. Section 4.3 introduces our method of query expansion using folksonomy. In Section 4.4, we discuss the different experiments that evaluate the performance of our approach. Finally, we conclude and provide some future directions in Section 4.5.

4.2 Problem Definition

As mentioned before, query expansion consists of enriching the initial query with additional information. This expansion is generally expected to provide better search results. However, providing merely a uniform expansion to all users is, from our point of view, not really suitable nor efficient since relevance of documents is user-dependent [PSC+02]. Thus, a simple and uniform query expansion is not enough to provide satisfactory search results for each user. Hence, having a folksonomy $\mathbb{F}(U, T, R)$, the problem we are addressing can be formalized as follows:

For a given user $u \in U$ who issued a query $q = \{t_1, t_2, ..., t_n\}$, how to provide for each $t_i \in q$ a ranked list of related terms $\mathcal{L} = \{t_{i1}, t_{i2}, ..., t_{ik}\}$ where $t_{ik} \in T$, such that when expanding the term t_i with the top k of $\mathcal{L}$, the most relevant results are provided to the user?

4.3 A New and Effective Personalized Social Query Expansion

The approach we are proposing aims at expanding user's queries in a personalized way. It can be decomposed into two parts: (i) *an offline* and (ii) *an online* part. The offline part performs the heavy computation which consists of transforming the whole social graph of folksonomy $\mathbb{F}$ into a graph of tags where two tags are related if they are semantically related. This part is also responsible for the construction and the update of the users' profiles, for serving the online part. The online part of the approach is responsible for computing the concrete expansion using the graph of tags and the user' profiles constructed in the offline part. In the following, we describe in more details each part and we explicitly highlight our contributions.

4.3.1 Offline Part

The offline part is also decomposed into two facets: (i) the transformation of the social graph of folksonomy $\mathbb{F}$ into a graph of tags, representing similarities between tags that either occur on the same resources or are shared by the same users, and (ii) the computation of the users' profiles to highlight their interests for personalizing their queries.

The approach is based on the creation and maintenance of a graph of tags that represents all the similarities that exist between the tags of the folksonomy $\mathbb{F}$. There exist two kinds of approaches that propose to achieve that: (i) the first kind is based on the co-occurrence of tags over resources, while (ii) the second kind is based on their co-occurrence over users. In the following, we detail these two approaches.

4.3.1.1 Extracting Semantics From Resources

In the first category of approaches [MCM$^+$09, JLS09, Mik07b], authors state that semantically related tags are expected to occur over the same resources. For example, tags that most occur for *google.com* on *delicious* are: *search, google, searchengine, engine, web, internet*.

Thus, extracting semantically related tags can be carried out by computing similarities. There exist many similarity measures [MCM$^+$09], but all of them need preprocessing that consists of reducing the dimensionality of the tripartite graph of a folksonomy $\mathbb{F}$ into a bipartite graph. This reduction is generally performed through aggregation methods. From the study of existing aggregation methods proposed in [MCM$^+$09], we have chosen the *projectional aggregation* along with the *Jaccard*, the *Dice*, and the *Overlap* similarity measures to compute the similarity between tags. We choose this aggregation method because its simplicity, and it is one which gives better results in semantic information extraction [MCM$^+$09]. Hence, we follow the same process as [MCM$^+$09] to extract a graph of related tags from a folksonomy $\mathbb{F}$

according to their co-occurrence over resources:

1. Using a function $\mathcal{F}$ on the whole *folksonomy* $\mathbb{F}$ performs a *projectional aggregation* over the user dimension, resulting in a bipartite graph Tag-Resource.
2. Then, using a function $\mathcal{G}$ on the resulting bipartite graph Tag-Resource provides a graph of tags $\mathcal{T}_R$, in which each link is weighted with the similarity between tags according to one of the similarity measures summarized in Table 4.1.

Table 4.1 – Summarization of similarity measures between tags.

Dice	$Dice(t_i, t_j) = 2 \times \frac{\lvert R_{t_i} \cap R_{t_j} \rvert}{\lvert R_{t_i} \rvert + \lvert R_{t_j} \rvert}$
Jaccard	$Jaccard(t_i, t_j) = \frac{\lvert R_{t_i} \cap R_{t_j} \rvert}{\lvert R_{t_i} \cup R_{t_j} \rvert}$
Overlap	$Overlap(t_i, t_j) = \frac{\lvert R_{t_i} \cap R_{t_j} \rvert}{min(\lvert R_{t_i} \rvert, \lvert R_{t_j} \rvert)}$

Therefore, we may obtain either a graph of tags $\mathcal{T}_R$ using the *Jaccard*, the *Dice*, or the *Overlap*. Note that we do not merge the similarity measures in a same graph of tags, meaning that a graph of tags is constructed using only one similarity measure.

We end-up with an undirected and weighted graph in which nodes represent tags, and an edge between two tags represents the fact that these tags occur together at least on one resource. The weights associated to the edges are computed from similarities between tags as explained beforehand. This first step is illustrated in the left upper part of Figure 4.2.

4.3.1.2 Extracting Semantics From Users

In the second category [Mik07b, BCK+08], authors state that correlated tags are also used by the same users to annotate resources. For example, the tags *Collaborative* and *Blog* whose the semantic meaning is similar have been used 13 557 times together by users in the *delicious* dataset studied in [WZB08].

This observation is more expected to happen in certain folksonomies, where users are encouraged to upload their personal resources which leads to generate private bookmarks, e.g. a folksonomy such as *CiteULike*, *Flickr*, or *YouTube* where users are expected to upload respectively their research papers, images, and videos. Therefore, similarly to the previous approach, [Mik07b] proposes to extract semantically related tags using the following process:

1. Using a function $\mathcal{G}'$ on the folksonomy $\mathbb{F}$ performs a *projectional aggregation* over the resource dimension for obtaining a bipartite graph Tag-User.
2. Then the function $\mathcal{F}'$ is used on the obtained Tag-User to get another graph of tags $\mathcal{T}_\mathcal{U}$ where similarities between tags are computed using one of the three previous similarity measures summarized in Table 4.1.

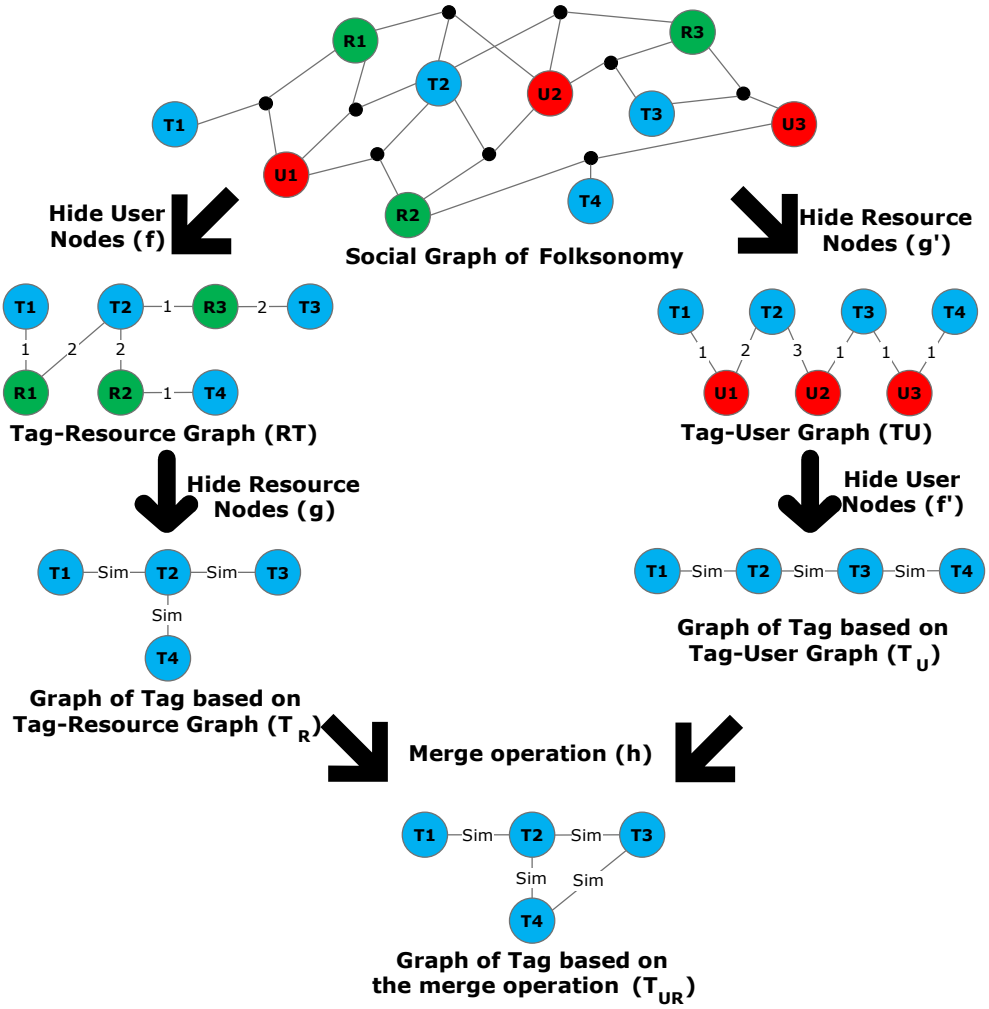


Figure 4.2 – Summary of the graph reduction process, which transform the whole social graph $\mathbb{F}$ into a graph of tag $\mathcal{T}_{\mathcal{UR}}$. The similarity values on the Figure are computed using the *Jaccard* measure on both graphs $\mathcal{T}_{\mathcal{R}}$ and $\mathcal{T}_{\mathcal{U}}$, and using $\alpha = 0.5$ on the graph $\mathcal{T}_{\mathcal{UR}}$.

This process is illustrated in the right upper part of Figure 4.2. Notice that the structure of the graph of tags $\mathcal{T}_{\mathcal{R}}$ is different from the one of the graph of tags $\mathcal{T}_{\mathcal{U}}$.

4.3.1.3 Construction of the Graph of Tag Similarities

Using only one of the two previous methods to construct a graph representing similarities between tags leads to a loss of information on one side or the other. For example, if we choose to extract related tags according to their co-occurrence over resources, we neglect the fact that there are some tags which are expected to be shared by the same users, and vice versa.

Therefore, we propose to use a function $\mathcal{M}$ which is applied on the graphs of tags $\mathcal{T}_{\mathcal{R}}$ and $\mathcal{T}_{\mathcal{U}}$ to merge them and to get a unique graph of tags $\mathcal{T}_{\mathcal{UR}}$ where the new

similarity values are computed by merging the values using the *Weighted Borda Fuse (WBF)* [DDR07]. This merge is computed as follows:

$$Sim_{\mathcal{T}_{\mathcal{UR}}}(t_i, t_j) = \alpha \times Sim_{\mathcal{T}_{\mathcal{R}}}(t_i, t_j) + (1 - \alpha) \times Sim_{\mathcal{T}_{\mathcal{U}}}(t_i, t_j) \tag{4.1}$$

Where $Sim_{\mathcal{T}_{\mathcal{UR}}}(t_i, t_j)$ calculates the similarity between two tags relying on the two other types of nodes, i.e. users and resources. The parameter α is a weight that satisfies $0 \leq \alpha \leq 1$, and represents the importance one wants to give to the two types of graphs, i.e. resources or users, in the consideration of the similarity calculation. In fact, depending on the context, when computing the similarity between two tags, one may want to give a higher importance to users sharing these two tags than documents having these tags as a common tags. Another user may want to give more importance to their co-occurrence over resources than to the users sharing these tags. Depending on the nature of the folksonomy, we set α to its optimal value in order to maximize the tags semantics extraction. Finally, it should be noted that this merge is performed between graphs generated with the same similarity measure.

Example 4.2. On *delicious*, tags are more expected to occur on resources because resources are commonly shared by users, i.e. α should be set to a high value. While on *flickr*, tags are more expected to be shared by users because resources are uploaded by each user to create personal bookmarks, i.e. α should be set to a low value.

Once the graph of tags $\mathcal{T}_{\mathcal{UR}}$ is constructed and the similarity values computed, we use Algorithm 4.1 to update the graph of tags $\mathcal{T}_{\mathcal{UR}}$.

In summary, if a new bookmark (u, t, r) is added to the folksonomy $\mathbb{F}$, we first check if the tag t already exists in the graph of tags $\mathcal{T}_{\mathcal{UR}}$ or not (line 1). If it exists, we have to get the instance of t (line 2), otherwise we add it to $\mathcal{T}_{\mathcal{UR}}$ (line 4). Next, we have to update the connection of the tag t inside the graph $\mathcal{T}_{\mathcal{UR}}$. To achieve this, we first get all the tags that occur over the resource r (line 6), then we link all these tags with t (line 8) and we update the new similarity values using one of the Equations of Table 4.1 (line 9). In the same way, we get all the tags that are used by the user u (line 11), and we link all these tags with t (line 13). Finally, we update the new similarity values using of the Equations of Table 4.1 (line 14).

The following example illustrates the computation of similarities and the graph transformation.

Example 4.3. Consider a folksonomy with three users $\{u_1, u_2, u_3\}$, three resources $\{r_1, r_2, r_3\}$, and four tags $\{t_1, t_2, t_3, t_4\}$ as illustrated in Figure 4.2. The ternary edges connecting the different entities represent bookmark instances (nine bookmarks in this example). We focus our example on the calculation of the similarity between the two tagst_1 and t_2. First, we use the function $\mathcal{F}$ to perform a distributional aggregation to obtain a bipartite graph Tag-Resource. Then, we compute the similarity between t_1 and t_2 in the graph $\mathcal{T}_R$ using of the Equations of Table 4.1 as follows:

Algorithm 4.1 Update graph of tags $\mathcal{T}_{\mathcal{UR}}$

Require: A folksonomy $\mathbb{F}$
An initial graph of tags $\mathcal{T}_{\mathcal{UR}}$
A new bookmark (u, t, r)

1: **if** t is already in $\mathcal{T}_{\mathcal{UR}}$ **then**
2: $t \leftarrow$ Retrieve the instance of t in $\mathcal{T}_{\mathcal{UR}}$
3: **else**
4: Add t to $\mathcal{T}_{\mathcal{UR}}$
5: **end if**
6: $l[n] \leftarrow$ list of n tags assigned to the ressource r in the folksonomy $\mathbb{F}$
7: **for all** $t_i \in l[n]$ **do**
8: Link t_i and t in $\mathcal{T}_{\mathcal{UR}}$
9: Update the similarity between t_i and t in $\mathcal{T}_{\mathcal{UR}}$ using one of the Equations of Table 4.1
10: **end for**
11: $l[m] \leftarrow$ list of n tags used by the user u in the folksonomy $\mathbb{F}$
12: **for all** $t_i \in l[m]$ **do**
13: Link t_i and t in $\mathcal{T}_{\mathcal{UR}}$
14: Update the similarity between t_i and t in $\mathcal{T}_{\mathcal{UR}}$ using one of the Equations of Table 4.1
15: **end for**

$Jaccard_{\mathcal{T}_{\mathcal{R}}}(t_1, t_2) = \frac{1}{3} = 0.33$, $Dice_{\mathcal{T}_{\mathcal{R}}}(t_1, t_2) = \frac{2}{1+3} = 0.5$, $Overlap_{\mathcal{T}_{\mathcal{R}}}(t_1, t_2) = \frac{1}{\min(1,3)} = 1$

Following these steps, we obtain a graph of tag similarities, which illustrated in the left side of Figure 4.2. Similarly, we use a function $\mathcal{G}'$ on $\mathbb{F}$ to perform a Distributional aggregation over resources, to obtain a bipartite graph Tag-User (the right side of Figure 4.2). Then, the function $\mathcal{F}'$ gives another graph of tags $\mathcal{T}_{\mathcal{U}}$ where we compute the similarity between t_1 and t_2 in the graph $\mathcal{T}_{\mathcal{U}}$ using again of the Equations of Table 4.1. As noticed before, the process is applied on all the other tags but we only use t_1 and t_2 for illustration.

$Jaccard_{\mathcal{T}_{\mathcal{U}}}(t_1, t_2) = \frac{1}{2} = 0.5$, $Dice_{\mathcal{T}_{\mathcal{U}}}(t_1, t_2) = \frac{2}{3} = 0.66$, $Overlap_{\mathcal{T}_{\mathcal{U}}}(t_1, t_2) = \frac{1}{\min(1,2)} = 1$

Finally, we use Formula 4.1 to merge the two values and to compute the similarities into the graph of tags $\mathcal{T}_{\mathcal{UR}}$ while fixing $\alpha = 0.5$ as follows:

$$Jaccard_{\mathcal{T}_{\mathcal{UR}}}(t_1, t_2) = 0.5 \times J_{\mathcal{T}_{\mathcal{R}}}(t_1, t_2) + 0.5 \times J_{\mathcal{T}_{\mathcal{U}}}(t_1, t_2) = 0.5 \times 0.33 + 0.5 \times 0.5 = 0.41$$

$$Dice_{\mathcal{T}_{\mathcal{UR}}}(t_1, t_2) = 0.5 \times D_{\mathcal{T}_{\mathcal{R}}}(t_1, t_2) + 0.5 \times D_{\mathcal{T}_{\mathcal{U}}}(t_1, t_2) = 0.5 \times 0.5 + 0.5 \times 0.66 = 0.58$$

$$Overlap_{\mathcal{T}_{\mathcal{UR}}}(t_1, t_2) = 0.5 \times O_{\mathcal{T}_{\mathcal{R}}}(t_1, t_2) + 0.5 \times O_{\mathcal{T}_{\mathcal{U}}}(t_1, t_2) = 0.5 \times 1 + 0.5 \times 1 = 1$$

Table 4.2 summarizes the similarities between tags of Figure 4.2 using our method, while fixing $\alpha = 0.5$.

Table 4.2 – Tag-Tag similarities matrix of the graph of Figure 4.2, while fixing $\alpha = 0.5$ (*J: Jaccard, D: Dice, O: Overlap*)

	t_1	t_2	t_3	t_4
t_1	1	J=0.41 D = 0.58 O = 1	0	0
t_2	J = 0.41 D = 0.58 O = 1	1	J = 0.33 D = 0.5 O = 0.75	J=0.16 D = 0.25 O = 0.5
t_3	0	J = 0.33 D = 0.5 O = 0.75	1	J = 0.25 D = 0.33 O = 0.5
t_4	0	J = 0.16 D = 0.25 O = 0.5	J = 0.25 D = 0.33 O = 0.5	1

This step of the offline part of our approach extracts semantics from the whole social graph of a Folksonomy $\mathbb{F}$ without loss of information, i.e. by exploiting the co-occurrence of tags over resources and users. This step leads to the creation of a graph of tags, where edges represent semantic relations between tags. This graph will be further used to extract terms that are semantically related to a given term of a query to perform the query expansion. The contribution at this stage is the combination of the graphs issued from resources and users to construct a better graph of tag similarities without loss of information. This is different from the existing approaches where only one graph is used.

In the following, we introduce our method of constructing and weighting the user profiles in order to personalize the expanded queries.

4.3.1.4 Construction of the User Profile

Our approach is expected to provide personalized expanded queries. To do so, we propose to build a user profile that consists of capturing information regarding real user interests. There are different ways to build user profiles [VCJ10, SAYMY08, HHHW08]. For example, a person can be modeled as a vector of attributes of his online personal profiles including the name, affiliation, and interests. Such simple factual data provides an inadequate description of the individual, as they are often incomplete, mostly subjective and do not reflect dynamic changes [HHHW08].

In the context of folksonomies, the user feedback is expected to be mostly explicit (because of the tagging action, where the user explicitly assigns tags to resources). For

example, adding a bookmark to my personal collection indicates my interests in the topic as well as the intention to access the page in the future. However, the fact that a bookmark is not in my collection does not necessarily represent a lack of interest (see Section 4.3.2.1 to see how we deal with this problem).

Thus, in a folksonomy, users are expected to tag and annotate resources that are interesting to them using tags that summarize their understanding of resources. In other words, these tags are in turn expected to be a good summary of the user's topics of interests as also discussed in [CZG+09, XBF+08, BXW+07, NM07, HHHW08, SCK+08]. Hence, each user can be modeled as a set of tags and their weights.

The definition of a user profile is given in Definition 2.8. The main challenge here is *how to define the weight of each dimension in the user profile?* We propose to use an adaptation of the well known *tf-idf* measure to estimate this weight. Hence, we define the weight w_{t_i} of the term t_i in the user profile as the *user term frequency, inverse user frequency (utf-iuf)*, which is computed as follows:

$$utf - iuf_{t_i,u_j} = \frac{n_{t_i,u_j}}{\sum_{t_k \in \vec{p_u^m}} n_{t_k,u_j}} \times log\left(\frac{|U|}{|U_{t_i}|}\right) \tag{4.2}$$

where n_{t_i,u_j} is the number of time the user u_j used the tag t_i.

A high value of *utf-iuf* is reached by a high user term frequency and a low user frequency of the term in the whole set of users. Note that we perform a stemming on tags before computing the profiles, to eliminate the differences between terms having the same root to better estimate the weight of each term.

User profiles are created offline and maintained incrementally. This is motivated by the fact that profiles and tagging actions are not evolving as quickly as query formulation on the system. As an analogy, it is well known that 90% of users in the social Web consume the content (i.e. query formulation), 9% update content, and 1% generate new content (profile updates) [Nie06]. Thus, we have decided to handle the profile construction as an offline task while providing a maintenance process for keeping it up to date.

In summary, at the end of the offline part, we build two assets: (i) a graph of tags similarities which is used to represent semantically relatedness of terms, and (ii) user profiles which will be leveraged in the personalization step.

4.3.2 Online Part

The online part of the approach is responsible for computing the concrete expansion using the graph of tags $\mathcal{T}_{\mathcal{UR}}$ and the user profiles constructed in the offline part. Before presenting our algorithm of query expansion, we propose a method to compute on the fly, the interest of a user to a given tag.

4.3.2.1 Interest Measure to Tag

Having computed the similarity graph between tags and built users' profiles containing the degree to which a set of tags are representative of a user, it becomes possible to compute a degree of interest a user may have to other tags, e.g. query tags. This is useful in our approach to compute, in real time, the suitable expansions of a tag w.r.t. a given user. In our approach, this interest is seen as a similarity between the user profile $\vec{p_u}$ and a tag t_i. Intuitively, the computed similarity captures the interest of the user u in the query term t_i denoted $\mathcal{I}_{t_i}^u$:

$$\mathcal{I}_u(t_i) = \sum_{t_j \in \vec{p_u}} (Sim_{\mathcal{T}_{\mathcal{UR}}}(t_i, t_j) \times w_j) \tag{4.3}$$

where $Sim(t_i, t_j)$ is the similarity between the term t_i and t_j, the j^{th} term of the user profile, and w_j is the weight of the term t_j in the profile computed during the previous process. Notice that any similarity measure can be used for computing $Sim(t_i, t_j)$, as discussed in [MCM+09]. In this work, we consider the *Jaccard*, the *Overlap*, and the *Dice* similarity measures, as discussed in the previous sections.

4.3.2.2 Effective Query Expansion

In this step of query expansion, we consider that the similarity between two terms t_i (a query term) and t_j (a potential candidate for the expansion of t_i), to be influenced by two main features:

1. The semantic similarity between t_i and t_j (the semantic strength between the two terms).
2. The extent to which the tag t_j is likely to be interesting to the considered user.

Once these two similarities are computed, a merge operation is necessary to obtain a final ranking value that indicates the similarity of t_j with t_i w.r.t. the user u. For this, several aggregation methods and algorithms exist. We choose the *Weighted Borda Fuse (WBF)* as summarized in Equation 4.4, where $0 \leq \gamma \leq 1$ is a parameter that controls the strength of the semantic and social parts of our approach. Using Equation 4.4, we can rank a list of terms $\mathcal{L}$, which are semantically related to a given term t_i from a user perspective.

$$Rank_t^u(t_j) = \overbrace{\gamma \times Sim_{\mathcal{T}_{\mathcal{UR}}}(t, t_j)}^{\texttt{Semantic Part}} + \underbrace{(1-\gamma) \times \mathcal{I}_{t_j}^u}_{\texttt{Social Part}} \tag{4.4}$$

The effective social query expansion is summarized in Algorithm 4.2. Hence, for a query $q = t_1 \wedge t_2 \wedge ... \wedge t_m$ issued by a user u, we first get the user's profile, which is computed as explained above (Section 4.3.1.4 and Line 1 in Algorithm 4.2). At this stage, the purpose is to enrich each term t_i of q with related terms (line 4). Then, the objective is to get all the neighboring tags t_j of t_i in the tag graph $\mathcal{T}_{\mathcal{UR}}$ (line 7).

After that (in line 4), we compute for each t_j, the ranking value that indicates its similarity with t_i w.r.t. the user u using formula 4.4 (line 5). Next, the neighbor list has to be sorted according to the computed values and we keep only the k top tags (line 7). Finally, t_i and its remaining neighbors must be linked with the OR ($\vee$) logical connector (line 8) and updated in q'.

Example 4.4. If a user issues a query $q = t_1 \wedge t_2 \wedge ... \wedge t_m$, it will be expanded to q'=$\{(t_1 \vee t_{11} \vee ... \vee t_{1l}) \wedge (t_2 \vee t_{21} \vee ... \vee t_{2k}) \wedge ... \wedge (t_m \vee t_{m1} \vee ... \vee t_{mr})\}$, where t_{ij} is a term that is semantically related to $t_i \in q$ and socially to u.

Algorithm 4.2 Effective Social Query Expansion

Require: A folksonomy $\mathbb{F}$
u : a User. $q = \{t_1, t_2, ..., t_n\}$: a Query.

1: $p_u[m] \leftarrow$ extract profile of u from $\mathbb{F}$
2: **for all** $t_i \in q$ **do**
3: $\mathcal{L} \leftarrow$ list of neighbor of t_i in tag graph $\mathcal{T}_{\mathcal{UR}}$
4: **for all** $t_j \in l$ **do**
5: $t_j.Value \leftarrow$ Compute the ranking score $Rank^u_{t_i}(t_j)$ as in equation 4.4
6: **end for**
7: Sort $\mathcal{L}$ according to $t_j.Value$ and keep only the top k terms in $\mathcal{L}$
8: Make a logical *OR* ($\vee$) connection between t_i and all terms of $\mathcal{L}$
9: Set the weight of the new terms t_j as the $t_j.Value$ or the TF-IDF value, depending on the choosed strategy (See Section 4.3.2.3)
10: Insert $\mathcal{L}$ in q'
11: **end for**
12: **return** q'

It should be noted that in this paper, we consider that the selection of each query term is determined independently, without considering latent term relations. Most past work on modeling term dependencies has analyzed three different underlying dependency assumptions: full independence, sequential dependence [SS02], and full dependence [MC05]. Taking into account terms dependency is part of our future works.

4.3.2.3 Terms Weighting

Term weighting in query expansion is challenging since there is no formal method for assigning weights to new terms. Indeed, appropriately weighting terms should result in better retrieval performance. Thus, we experiment the following two strategies for weighting new terms:

1. Using the ranking values of Formula 4.4 as the weight of the new expanded terms. This strategy provides personalized term weight assignment while considering both semantic strength and user interest.

2. Using the *Term Frequency-Inverse Document Frequency (TF-IDF)* [BYRN11] as the weight of the new expanded terms as follows:

$$tf - idf_{t_i,q} = tf_{t_i} \times log\left(\frac{|D|}{|D_{t_i}|}\right) \tag{4.5}$$

where tf_{t_i} denotes the term frequency of t_i in the query q. This strategy provides a uniform term weight to the query while keeping the personalizing aspect in choosing terms. Notice that weights are assigned to terms in the line 9 of Algorithm 4.2.

4.3.3 Complexity Analysis

The main strength of our approach is its ability to provide a fast personalized query expansion according to each user, while taking into account his last interactions in the social network. In each iteration, the time complexity of the algorithm for computing the ranking values is $O(m \times n)$ (m is the profile length and n is the number of neighbors of each term t_i of the query) and the complexity for ranking terms according to the computed values is $O(n \times log(n))$ (this corresponds to the complexity of sorting an array of n elements). Therefore, the total computational complexity in one iteration is:

$$O(n \times log(n) + m \times n)$$

However, the distribution of tags over resources and users in folksonomies follows a power law [SBC+10, WZB08, HKGM08]: most URLs are tagged by only a handful of users, and many tags are only used by a few users, a property which leads to a low values of n and m. Therefore, our algorithm has a low complexity, which makes it very efficient and scale to very large datasets.

4.4 Evaluations

In this section, we describe the two types of evaluations we performed on our approach:

1. An estimation of the parameters of our approach to provide insights regarding their potential impact on the system.
2. A comparison study, where our approach is compared to the closest state of the art approaches to provide insights about the obtained results and position the proposal.

In the following, we provide a description of the used datasets and the evaluation methodology.

4.4.1 Datasets

A number of social bookmarking systems exist [HHLS05]. We have selected three datasets to perform an offline evaluation: *delicious, flickr and CiteULike*. These datasets are available and public. The interest of using such data instead of crawled data is to work on widely accepted data sets, reduce the risk of noise, and possible reproduce the evaluations by others as well as the ability to compare our approach to other approaches on "standardized datasets". Hereafter is the description of the different datasets.

— **delicious:** a social bookmarking web service for storing, sharing, and discovering web bookmarks. We have used a dataset which is described and analyzed in [WZB08] [26].

— **Flickr:** an image hosting, tagging and sharing website. The *Flickr* dataset is the one used and studied in [SBC+10] [27].

— **CiteULike:** an online bookmarking service that allows users to bookmark academic papers. This dataset is the one provided by the *CiteULike* website [28].

Before the experiments, we performed three data preprocessing tasks: (1) Several annotations are too personal or meaningless, such as "toread", "Imported IE Favorites", "system:imported", etc. We remove some of them manually. (2) Although the annotations from *delicious* are easy for users to read and understand, they are not designed for machine use. For example, some users may concatenate several words to form an annotation such as "java.programming" or "java/programming". We split this kind of annotations before using them in the experiments. (3) The list of terms undergoes a stemming by means of the Porter's algorithm [Por97] in such a way to eliminate the differences between terms having the same root. In the same time, the system records the relations between stemmed terms and original terms. As for the *delicious* dataset, we add two other data preprocessing tasks:

1. We downloaded all the available web pages while removing those which are no longer available.
2. We removed all the non-english web pages. This operation was performed using *Apache Tika* toolkit.

Table 4.3 gives a description of these datasets.

4.4.2 Evaluation Methodology

Making evaluations for personalized search is a challenge in itself since relevance judgements can only be assessed by end-users themselves [CZG+09]. This is difficult to achieve at a large scale. Different efforts [KHS08, BHJ+10, BFNP08, MJSZ07] state

26. http://data.dai-labor.de/corpus/delicious/
27. http://www.tagora-project.eu/data/#flickrphotos
28. http://static.citeulike.org/data/2007-05-30.bz2

Table 4.3 – Corpus details

	Bookmarks	Users	Resources	Tags
delicious	9 675 294	318 769	425 183	1 321 039
Flickr	22 140 211	112 033	327 188	912 102
CiteULike	16 164 802	107 066	3 508 847	712 912

that the tagging behavior of a user of folksonomies closely reflects his behavior of search on the Web. In other words, if a user tags a resource r with a tag t, he will choose to access the resource r if it appears in the result obtained by submitting t as query to the search engine. Thus, we can easily state that any bookmark (u, t, r) can be used as a test query for evaluations. The main idea of the experiments is based on the following assumption:

Assumption 4.4.1. *For a personalized query $q = \{t\}$ issued by user u with query term t, the relevant documents are those tagged by u with t.*

Hence, in the off-line study, for each evaluation, we randomly select 2000 pairs (u, t), which are considered to form a personalized query set. For each corresponding pair (u, t), we remove all the bookmarks $(u, t, r) \in \mathbb{F}, \forall r \in R$ in order to not promote the resource r in the obtained results. For each pair, the user u sends the query $q = \{t\}$ to the system. Then, the query q is enriched and transformed into q' following our approach. For the *delicious* dataset, documents that match q' are retrieved, ranked and sorted using the *Apache Lucene*. For the *Flickr* and *CiteULike* datasets, we retrieve all resources that are annotated with tags of q while representing them according to the *Vector Space Model (VSM)* (See Section 2.2.2).

For the *Flickr* and *CiteULike* datasets, we rank all the retrieved resources using values of the cosine similarity and we consider that relevant resources are those tagged by u using tags of q' to assess the obtained results. The random selection was carried out 10 times independently, and we report the average results.

As a last step, fro each obtained list of results, we compute the Mean Average Precision (see Equation 2.4) and the Mean Reciprocal Rank (see Equation 2.5) over the 2000 queries.

4.4.3 Study of the Parameters

We intend here to observe the parameters of our approach and estimate their optimal values. These parameters are:

1. γ, which controls the semantic part and the social part in the ranking of tags for an expansion (see Equation 4.4). The higher its value is, the stronger is the semantic part in tag similarity ranking, and vice versa.
2. The number of tags which are suitable for the expansion.

3. α, which gives either a higher importance to resources or to users, when computing the graph of tags $\mathcal{T}_{\mathcal{UR}}$. We set this parameter such that: the higher its value is, the stronger are the resources' links, and thus weaker the users links are, and vice versa (see Equation 4.1).
4. We evaluate two strategies for weighting the expanded terms (see Section 4.3.2.3).
5. Finally, we observe the impact of the similarity measures over the search results.

We refer to our approach in Figures 4.3, 4.4, 4.5, and 4.6 as *Personalized Social Query Expansion (PSQE)*. Also, all the Figures contain the results according to each similarity measure, and for each similarity measure, the results of the two weighting strategies are shown (this results in six curves per graph).

4.4.3.1 Impact of the Social Interest (γ)

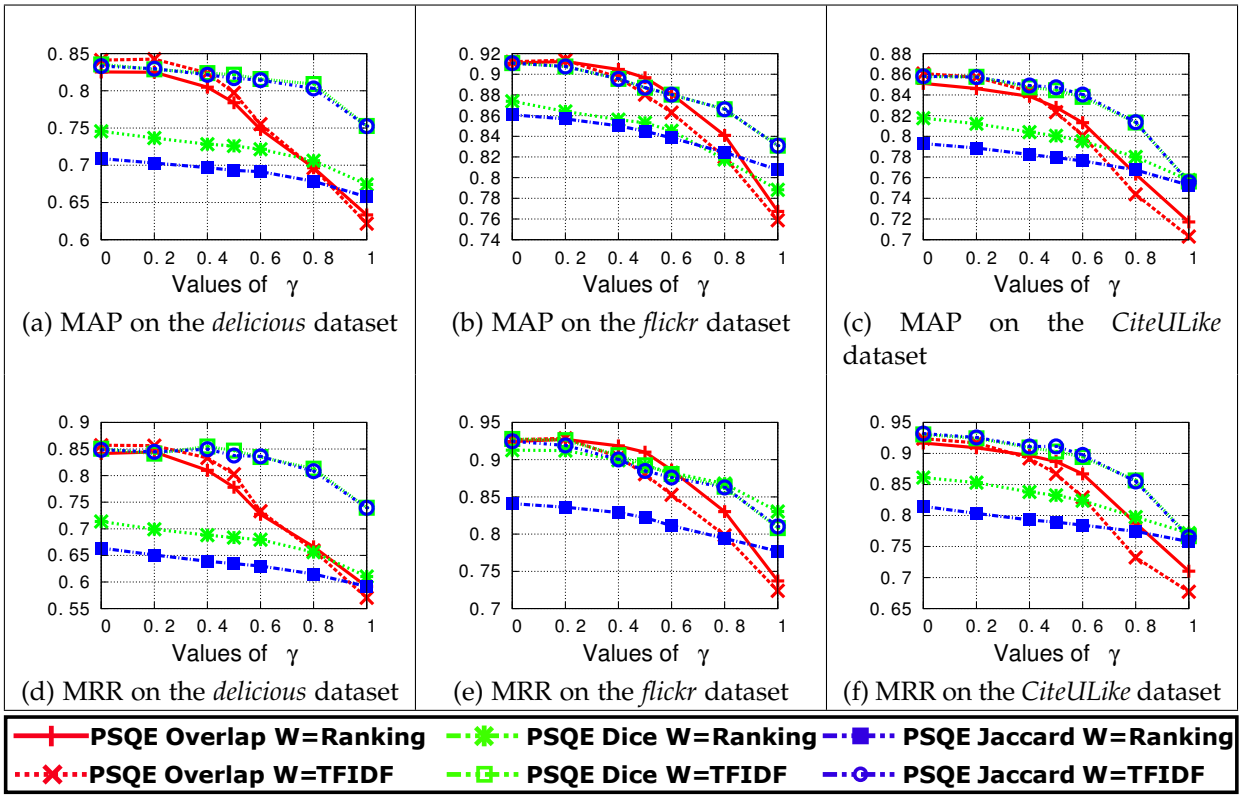
The results showing the impact of the user interest regarding the semantic similarity is given in Figure 4.3. This latter shows the evolution of the MAP and the MRR for different values of γ, while fixing $\alpha = 0.5$ and query size to 4 for our three datasets, and using the three similarity measures. We note that the smaller the value of γ is, the better is the performance. This can be explained by the fact that the higher the value of the user interest part, the more resources that the user tags are highlighted (probably other users tag them with the same tags), and the higher is the value of the MAP and the MRR. However, we consider that neglecting the semantic part of Equation 4.4 is not suitable for the following reasons:

1. First, if we fix γ to 0, we are going to neglect the semantic part, and perhaps lose the query sense (even if the potential terms to expand the query are those related to the query terms);
2. Second, if we fix γ to 0 we are going to face cold start problems, since new users don't have an initial profile that allows us to rank terms.

Thus, we choose to fix γ to 0.5 for the rest of the evaluations.

4.4.3.2 Impact of the Query Size

The objective here is to check if the length of a query impacts the obtained results. The results are illustrated in Figure 4.4. Through all the experiments we have performed, it comes out that the maximum performance is achieved while adding 4 to 6 related terms to the query. Adding more than 6 related terms has no impact on the quality of the results when using values of Formula 4.4 as weight for new term. This has even a negative impact when using TF-IDF values for term weighting as Figure 4.4 shows. For the first case, this is due to the fact that the weight of the added terms is close to 0 (we remind that the weight of the added terms is the value of Formula 4.4). Hence, this makes it natural and intuitive to pick a value in the provided interval, between 4 and 6.



(a) MAP on the *delicious* dataset (b) MAP on the *flickr* dataset (c) MAP on the *CiteULike* dataset

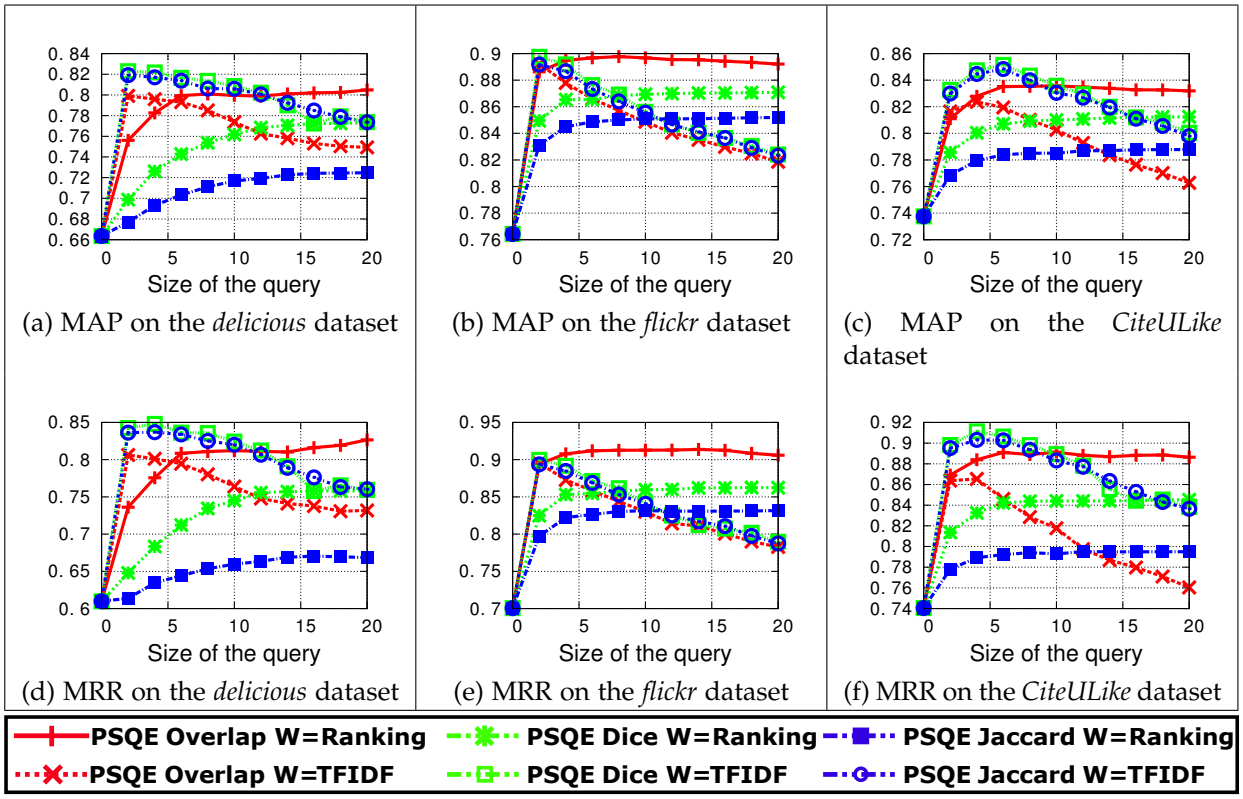
(d) MRR on the *delicious* dataset (e) MRR on the *flickr* dataset (f) MRR on the *CiteULike* dataset

Figure 4.3 – Measuring the impact of the social interest (γ). For different values of γ, we fix $\alpha = 0.5$, query size=4 and we use the three similarity measures and the two weighting strategies for new terms averaged over 1000 queries, using the VSM.

4.4.3.3 Impact of the Users and Resources (α)

The importance of users and resources on the way the expansion is performed can be tuned by the parameter α of Equation 4.1. Fixing $\alpha = 0$ considers only links between tags based on common users while fixing $\alpha = 1$ considers only links between tags based on common resources. The results regarding this parameter are illustrated in Figure 4.5, where the MAP and the MRR's behaviors are quite different on the three datasets.

Indeed, in the *delicious* dataset, the values of the MAP and MRR increase by increasing the value of α using both the *Jaccard* and the *Dice* similarities achieving an optimal performance at $\alpha = 1$. As for *Flickr* and *CiteULike*, the optimal performance is achieved for $\alpha = 0.2$ and $\alpha = 0.5$ respectively. We believe that this is due to the fact that in social bookmarking systems like *delicious*, users are expected to share and annotate the same resources (URLs in *delicious*) to give rise to less private resources. Therefore, annotations are expected to occur more on resources than on users. However, in social bookmarking systems like *Flickr* and *CiteULike*, users are expected to upload their own resources (images and papers) resulting in more private resources. Thus, annotations are expected to occur more on users than on resources, a property



(a) MAP on the *delicious* dataset (b) MAP on the *flickr* dataset (c) MAP on the *CiteULike* dataset

(d) MRR on the *delicious* dataset (e) MRR on the *flickr* dataset (f) MRR on the *CiteULike* dataset

Figure 4.4 – Evaluating the impact of the query size on the expansion. For different values of the query size, we use $\gamma = 0.5$, $\alpha = 0.5$ and our two strategies of weighting new terms.

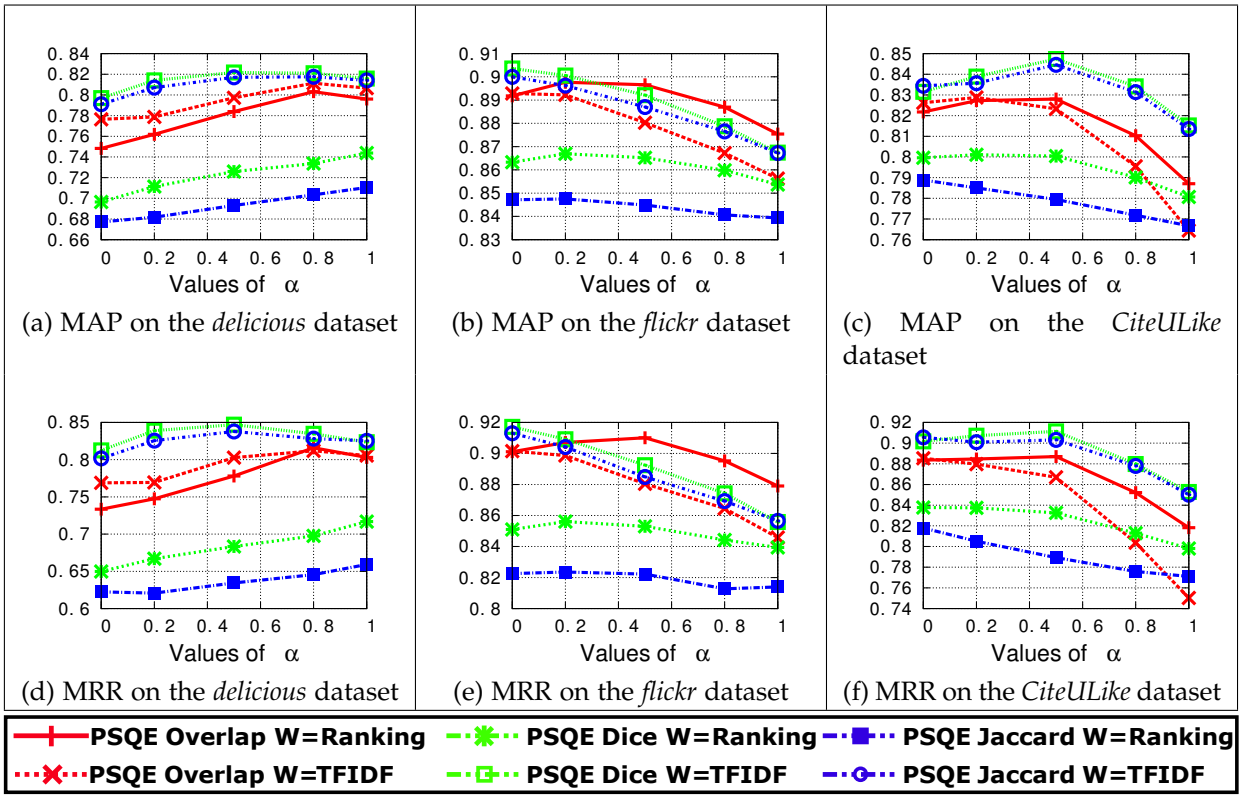
which has been also observed and reported in [CRYT10].

4.4.3.4 Impact of the Weight of Terms

In Section 4.3.2.3, we explain that we experiment two strategies for weighting the new expanded terms by either (i) using value of Formula 4.4, or (ii) the *TF-IDF* value using Formula 4.5. We note that the performances follow almost the same distribution while varying γ and α in Figure 4.3 and 4.5, and for our three similarity measures over our three datasets. However, we report that each time, the *TF-IDF* weighting strategy provides better performance. Hence, we conclude that personalizing the term weighting is less advantageous and less efficient comparing to a uniform weighting approach as used in the second strategy.

4.4.3.5 Impact of the Similarity Measures

The behavior of the performance seem to be the same for the three similarity measures with each time a small advantage to the *Dice* measure. Hence, taking into account the ratio between all the entities to which two tags are associated together



(a) MAP on the *delicious* dataset (b) MAP on the *flickr* dataset (c) MAP on the *CiteULike* dataset

(d) MRR on the *delicious* dataset (e) MRR on the *flickr* dataset (f) MRR on the *CiteULike* dataset

Figure 4.5 – Evaluating the impact of the users/resources on the expansion. For values of α, using the three similarity measures, $\gamma = 0.5$, query size=4 and for our two strategies of weighting new terms.

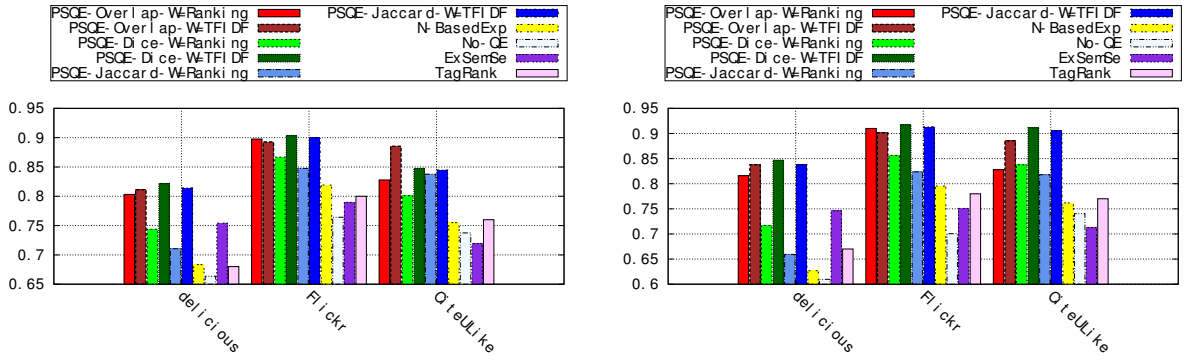
versus the union of these entities leads to a better estimation of the similarity in folksonomies.

4.4.4 Comparison With Existing Approaches

Our objective here is to estimate how well our approach meets the users' information needs and compare its retrieval quality to that of other approaches, objectively. Our approach is evaluated using the optimal values computed in the previous section and using our two strategies of term weighting as explained in Section 4.3.2.3. The results are illustrated in Figure 4.6 as *"PSQE-W = Ranking"* for the first strategy and *"PSQE-W=TFIDF"* for the second strategy, where we select four baselines for comparison as described in the following. Note that we choose the parameters that give the optimal performance for each of these baselines.

4.4.4.1 PSQE vs NoQE

The first approach for comparison is that with no query expansion or personalization. Documents that match queries are retrieved, and ranked as explained above. We



(a) Mean Average Precision (MAP)

(b) Mean Reciprocal Rank (MRR)

Figure 4.6 – Comparison with the different baselines of the MAP and MRR, while fixing $\gamma = 0.5$ and query size=4, using the *delicious*, *Flickr*, and *CiteULike* datasets. We choose the optimal value of α for each similarity measure.

report the following improvements:

— On the *delicious* dataset, we obtain an improvement of almost 13% of the MAP and 18% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 16% of the MAP and 24% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *flickr* dataset, we obtain an improvement of almost 13% of the MAP and 21% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 14% of the MAP and 21% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *CiteULike* dataset, we obtain an improvement of almost 10% of the MAP and 7% of the MRR for our first strategy of term weighting using the Jaccard similarity measure, and an improvement of almost 15% of the MAP and 14% of the MRR for our second strategy of term weighting using the Overlap similarity measure.

Thus, it is clear that the query expansion has an evident advantage compared to a strategy with no expansion. We refer to this approach as **NoQE** in Figure 4.6.

4.4.4.2 PSQE vs N-BasedExp

The second approach is the neighborhood based approach, which is based on the co-occurrence of terms over resources. This approach consists of enriching the query q with the most related terms without considering the user profile. Thus, queries are enriched similarly for each user. Our approach significantly outperform the neighborhood based approach as follows:

— On the *delicious* dataset, we obtain an improvement of almost 12% of the MAP and 19% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 14% of the MAP and 22% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *flickr* dataset, we obtain an improvement of almost 8% of the MAP and 12% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 9% of the MAP and 12% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *CiteULike* dataset, we obtain an improvement of almost 8% of the MAP and 5% of the MRR for our first strategy of term weighting using the Jaccard similarity measure, and an improvement of almost 13% of the MAP and 12% of the MRR for our second strategy of term weighting using the Overlap similarity measure.

Therefore, we conclude that our personalized query expansion efforts bring a considerable contribution according to an approach based on the most related terms. We refer to this approach as **N-BasedExp** in Figure 4.6.

4.4.4.3 PSQE vs ExSemSe

The third approach is an approach proposed in [BCK+08], which is a strategy that uses semantic search with query expansion named *Expanded Semantic Search*. In summary, this strategy consists of adding to the query q, k possible expansion tags with the largest similarity to the original tags in order to enrich its results. For each query, the query initiator u, ranks results using BM25 and tag similarity scores. We implemented this strategy and evaluated it over our datasets. We refer to this approach as **ExSemSe** in Figure 4.6. We report the following improvements:

— On the *delicious* dataset, we obtain an improvement of almost 5% of the MAP and 7% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 7% of the MAP and 10% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *flickr* dataset, we obtain an improvement of almost 11% of the MAP and 16% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 12% of the MAP and 16% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *CiteULike* dataset, we obtain an improvement of almost 12% of the MAP and 10% of the MRR for our first strategy of term weighting using the Jaccard similarity measure, and an improvement of almost 17% of the MAP and 17% of

the MRR for our second strategy of term weighting using the Overlap similarity measure.

4.4.4.4 PSQE vs TagRank

The fourth approach is an approach proposed in [BGLK09], which is an algorithm called *TagRank* that automatically determines which tags best expand a list of tags in a given query. We implemented this strategy and evaluated it over our datasets. We refer to this approach as **TagRank** in Figure 4.6. We report the following improvements:

— On the *delicious* dataset, we obtain an improvement of almost 18, 10% of the MAP and 21, 79% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 20, 83% of the MAP and 26, 42% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *flickr* dataset, we obtain an improvement of almost 12, 20% of the MAP and 16, 67% of the MRR for our first strategy of term weighting using the Overlap similarity measure, and an improvement of almost 12, 94% of the MAP and 17, 58% of the MRR for our second strategy of term weighting using the Dice similarity measure.

— On the *CiteULike* dataset, we obtain an improvement of almost 10, 23% of the MAP and 8, 79% of the MRR for our first strategy of term weighting using the Jaccard similarity measure, and an improvement of almost 16, 49% of the MAP and 18, 35% of the MRR for our second strategy of term weighting using the Overlap similarity measure.

In summary, the obtained results show that our approach of personalization in query expansion using social knowledge may significantly improve web search. By comparing the PSQE framework to the closest state of the art approaches, we show that it is a very competitive approach that mays provide high quality results whatever the dataset used. Finally, we notice that the better performance are obtained with the *Dice* similarity measure and using TF-IDF for term weighting over our three datasets.

4.5 Conclusion and Future Work

This Chapter discusses a contribution to the area of query expansion leveraging the social context of the Web. We proposed a new approach based on social personalization to transform an initial query q to another query q' enriched with close terms that are mostly used by not only a given user but also by his social relatives. Given a social graph (folksonomy), the proposed approach starts by creating and maintaining a similarity graph of tags, that represents semantic strength between tags. The steps required to generate this graph of tags is operated offline, before the system is ready

to process any query. Once this graph is created, a user profile is also created offline and maintained for each user online. These structures are used to compute personalized expansions on the fly thanks to the combination of the semantic and social dimensions. Finally, a formal evaluation of the quality of the results, using datasets crawled from *delicious, Flickr, and CiteULike* has shown the benefits of this approach in comparison to other approaches.

Even with the interest of the proposed method, there are still possible improvements that one can bring. We believe that our approach is complementary to some existing approaches in the area of SIR. Thus, we are convinced that a combination with social ranking functions such as those proposed in [CZG+09, XBF+08, NM07, HJSS06b] can be of a great interest.

Chapter 5

Using Social Annotations to Enhance Document Representation for Personalized Search

This chapter introduces our contribution to IR modeling. Since each user has his own understanding and point of view of a given document, we propose a Personalized Social Document Representation (PSDR) of each document per user based on his social activities. The proposed approach relies on matrix factorization to compute the PSDR of documents that potentially match the query terms at query time. The complexity analysis shows that our approach scales linearly with the number of documents that potentially match the query and can thus be applied to very large datasets.

5.1 Introduction

Modeling in IR consists of two main tasks [BYRN11]: (i) the definition of a *conceptual model* to represent documents [29] and queries and (ii) the definition of a ranking function to quantify the similarities among documents and queries. In this Chapter, we propose a contribution to IR modeling, motivated mainly by the following observations:

1. "Social contextual summarization" is required due to the fact that with the advent of the social Web where all users are contributors, web pages are associated to a social context that can tell us about their content (e.g social annotations). Hence, several research works ([ZYW+09, CRYT10, DEFS06, BFNP08]) reported that adding a tag to the content of a document enhances the search quality as they are good summaries for documents [BFNP08] (e.g. document expansion [SP99, HKGM08]). In particular, social information can be useful for documents that contain few terms where a simple indexing strategy is not expected

29. We also refer to documents as web pages or resources.

to provide a good retrieval performances (e.g. the *Google homepage* [30]).

2. "Common collaborative vocabularies" are needed to support common understanding since given a document, each user has his own understanding of its content. Therefore, each user uses a different vocabulary and words to describe, comment, and annotate this document. For example, if we look at the homepage of Youtube [31], a given user can tag it using "video", "Web" and "music" while another can tags it using "news", "movie", and "media".
3. "Relevance relativeness" is needed since relevance is actually relative for each user [PSC+02]. Hence, adapting search results according to each user in the ranking process is expected to provide good retrieval performance.

Motivated by these observations, we believe that enhancing the representation of documents and personalizing them with social information is expected to improve web search.

The approach we are proposing relies on users annotations as source of social information, which are associated to documents in bookmarking systems. As illustrated in Figure 5.1, the textual content of a document is shared between users under a common representation, i.e. all terms in a document are identically shared and presented to users as in a classic IR model, while the annotations given by a user to this document express his personal understanding of its content. Thus, these annotations symbolize a personal representation of this document to this user, e.g. the red annotations given by *Bob* to the document express his personal representation of this document, while green annotations constitute the personal representation of this document to Alice since she used them to describe its content. In this Chapter, our main objective is to answer the following question: *How to formalize a personal representation of a document in a social collaborative setting, and how to improve web search while relying on this representation?*

The main contributions of this chapter can be summarized as follows:

1. A document representation called Personalized Social Document Representation (PSDR), which is based on social information that comes from social bookmarking systems. The PSDR is expected to deliver, for a given document, different social representations according to each user based on the feedback of other users.
2. A key problem in an IR model is the definition of a ranking function used to establish a simple ordering of the documents retrieved. Hence, we propose two ranking functions that take into account both the textual content of documents and their PSDR according to the query issuer.

30. http://www.google.com/
There are only a very few terms on the page itself but a thousands of annotations available on *delicious* are associated to it. Eventually, the social information of the *Google* homepage are more useful for indexing.
31. http://www.youtube.com/

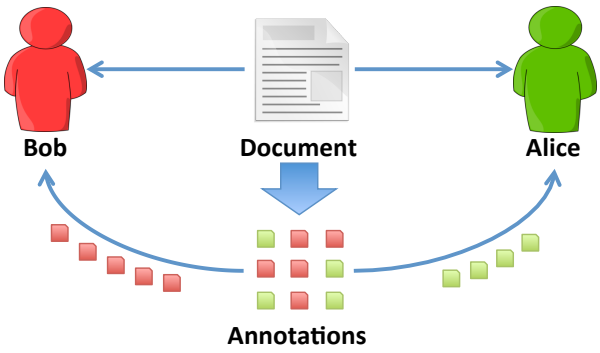


Figure 5.1 – Document representations for two users.

3. Our approach is validated via an intensive evaluation study on a large public dataset [WZB08]. This shows to which extent our approach contribute to an efficient Web search at the expense of existing approaches. The complexity analysis shows that our approach can be applied to large datasets since it scales linearly with the number of documents that match the query.

This Chapter is organized as follows: in Section 5.2, we introduces our approach of Personalized Social Document Representation and our ranking functions. In Sections 5.3, 5.4, and 5.5, we discuss the different experiments that evaluate the performances of our approach. Finally, we conclude and provide some future directions in Section 5.7.

5.2 Personalized Social Document Representation

In this section, we first illustrate our approach using a simple toy example. Then we introduce our method of modeling a social representation of document as a PSDR. Finally, we show how to use a PSDR for ranking purpose.

5.2.1 Toy example and Approach Overview

Before diving into the details of our approach, we describe hereafter an indicative scenario to illustrate our proposal throughout this chapter.

Example 5.1. Suppose that a user *Bob* issues the query *"news on the web"* to which a number of web pages that potentially correspond are retrieved. Let consider the web page *YouTube.com* as a document that matches this query. This web page is associated to many bookmarks in a folksonomy as illustrated in Figure 5.2. There are eight users (*Alice, Bob, Carol, Eve, Mallory, Nestor, Oscar, and Trudy*) who annotated *YouTube.com* using seven tags (info, web, video, news, blog, social, and mine).

Our approach intends to create a representation for each of these retrieved web pages from the perspective of *Bob*, on the fly, based on their social annotations. These

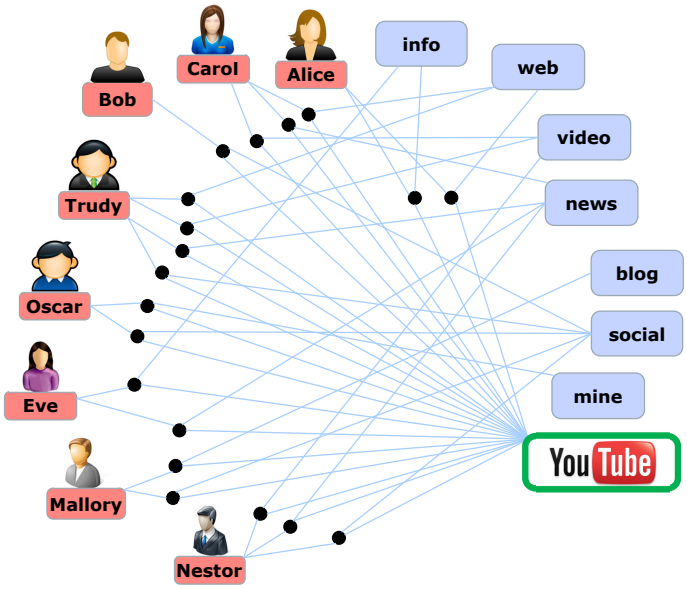


Figure 5.2 – Example of a folksonomy with eight users who annotate one resource using seven tags. The triples (u, t, r) are represented as ternary-edges connecting a user, a resource and a tag.

representations are used in order to compute a ranking score w.r.t. the query. Since this representation is specific to *Bob*, it is by definition personalized and we call it from now on, Personalized Social Document Representation (PSDR). For a given web page (e.g. *YouTube.com*), the only consideration of the user's tags as his personalized representation will result either in: (i) ignoring this web page if he did not annotate it (a user doesn't tag all web pages) or (ii) assigning it an inappropriate ranking score (since the representation is only based on his own perspective, it may be poor). Our goal is then to use other users' annotations to enrich the personalized representation of the query issuer enabling him to: (i) benefit from others' experiences and feedbacks, (ii) promote used/visited resources even if they are not well classified, and (iii) discover new resources.

For a document that potentially matches a query, our method proceeds into three main phases in order to collect as much useful information about this document and its social relatives. This information is reused to create its PSDR according to the query issuer. These phases are the following, as illustrated in Figure 5.3:

1. Representing each document that matches the query terms using a Users-Tags matrix. This matrix is first sized by selecting relevant users, e.g. *Carol, Nestor and Alice*, then it is weighted for estimating the extent to which each user thinks that a tag is associated to the considered document, e.g. *Alice* thinks that *info* is associated to *youtube.com* with a weight of 0.5. This phase includes four steps enumerated from 1 to 4 in Figure 5.3.
2. Each row i in a Users-Tags matrix of a given document represents the personal representation of the user u_i. This matrix is expected to be sparse, since it con-

tains many missing values that should be inferred to enhance the PSDR. Hence, a matrix factorization process is used to infer the PSDR of the considered document to the query issuer based on identifying weighting patterns. This phase corresponds to step 5 in Figure 5.3.

3. Finally, ranking documents based on their PDSR and their textual content. This phase is illustrated in steps 6 and 7 in Figure 5.3.

We detail in the following these different phases illustrated with our toy example.

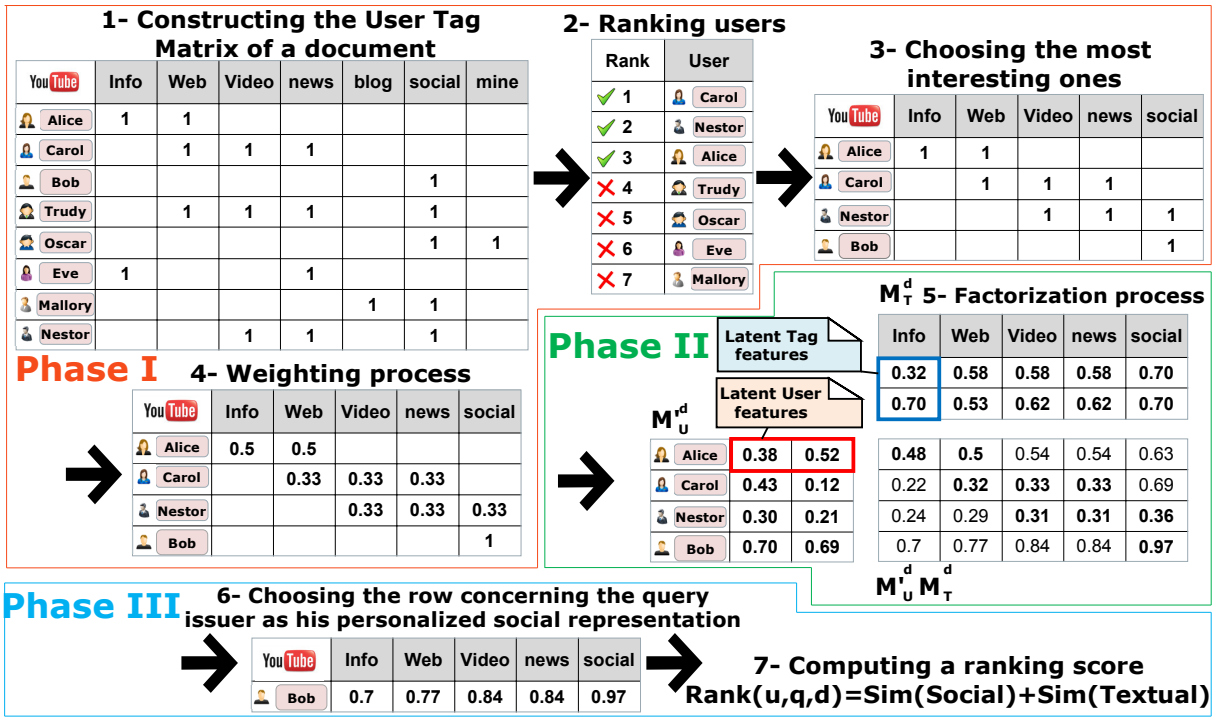


Figure 5.3 – Process of creating a personalized social representation of the web page *YouTube.com* to the user *Bob* of the folksonomy of Figure 5.2.

5.2.2 Constructing the Users-Tags Matrix

We detail here how we represent a web page using a Users-Tags matrix, and how it is weighted. This matrix will be subsequently used to infer the PSDR of the considered web page according to a query issuer.

5.2.2.1 Sizing the Users-Tags Matrix

The objective in this first step is to gather as much useful information as possible around the user and the relatives who may serve to construct and enrich the PSDR. As illustrated in Figure 5.3, each web page can be represented using an $m \times n$ Users-Tags matrix $M^d_{U,T}$ of m users who annotate the web page and the n tags that they used to

annotate it. Each entry w_{ij} in the matrix represents the number of times the user u_i used the term t_j to annotate the considered web page.

Example 5.2. In the folksonomy of Figure 5.2, *Bob* used the term *video* to annotate the web page *Youtube.com* once. A stemming is performed over terms before building the User-Tag matrix. Hence, if a user uses the terms *new* and *news* to annotate a web page, we consider only the term *new*, and we put the value 2 in the cell that corresponds to this user and this term when building the matrix.

Instead of using all users' feedback to infer a PSDR of the considered web page to *Bob*, we propose to choose only the most representative ones in order to filter out irrelevant users who may represent noise. To do so, we use a ranking function to rank users from the most relevant to the less relevant ones, and select only the top k users as the most representative to both the query issuer and the considered web page (see Step 2 of Figure 5.3). The irrelevant users may:

1. have annotated improperly a lot of documents;
2. have annotated the considered document with few terms;
3. not be socially close to the query issuer and don't share the same topics of interests.

Then, we select only the terms that the top k users employed to annotate this web page and build a new reduced Users-Tags matrix, which is expected to be more representative to both the query issuer and the considered web page (see Step 3 in Figure 5.3). Note that even if the query issuer has annotated the considered web page, we do not consider him in the ranking process since we want to rank users with respect to him.

At this stage, we assume that the ranking score of a user u according to a document d and the query issuer u_q takes into account: (i) the proximity of u to d, and (ii) the social proximity of u to u_q.

On the one hand, we propose to compute the proximity between the user u and the document d as an adaptation of the well known *tf-idf* measure. It is computed by merging the tagging action frequency of u in d, and the tagging behavior of u on d as follows:

$$Proximity(u,d) = \frac{\mid T_{u,d} \mid}{\mid T_d \mid} \times log\left(\frac{\mid D \mid}{\mid D_u \mid}\right) \tag{5.1}$$

A high proximity between u and d is reached by a high tagging action frequency (of u in d) and a low tagging action frequency of u in the whole collection of documents.

On the other hand, the social proximity between the user u and the query issuer u_q is computed as a similarity between their used tags. As described in [MCM+09], the similarity between two users can be computed using one of the measures mentioned in Table 5.1.

Finally, the ranking score of a user u according to a document d and the query issuer u_q is computed by merging the two previous scores as follows:

Table 5.1 – Summarization of similarity measures between users (i.e. $Sim(u, u_q)$).

Dice	$Dice(u, u_q) = 2 \times \frac{\lvert T_u \cap T_{u_q} \rvert}{\lvert T_u \rvert + \lvert T_{u_q} \rvert}$
Jaccard	$Jacard(u, u_q) = \frac{\lvert T_u \cap T_{u_q} \rvert}{\lvert T_u \cup T_{u_q} \rvert}$
Overlap	$Overlap(u, u_q) = \frac{\lvert T_u \cap T_{u_q} \rvert}{min(\lvert T_u \rvert, \lvert T_{u_q} \rvert)}$
Cosine	$Cos(u, u_q) = \frac{\overrightarrow{P_u} \bullet \overrightarrow{P_{u_q}}}{\lvert \overrightarrow{P_u} \rvert \times \lvert \overrightarrow{P_{u_q}} \rvert}$

$$Rank^d_{u_q}(u) = \overbrace{\alpha \times \frac{\mid T_{u,d} \mid}{\mid T_d \mid} \times log\left(\frac{\mid D \mid}{\mid D_u \mid}\right)}^{\texttt{Proximity to the document}} + \underbrace{(1-\alpha) \times Sim(u, u_q)}_{\texttt{Proximity to the query issuer}} \tag{5.2}$$

where α is a weight that satisfies $0 \leq \alpha \leq 1$, which allows giving more importance to either the document proximity part or to the query issuer proximity part.

Once we get a ranked list of users using Equation 5.2, we select the top k as the most representative ones to both the considered document and the query issuer. Then, we select their tags to built a new (smaller) Users-Tags matrix $M^d_{U,T}$. Finally, we add the query issuer as a new entry in the Users-Tags matrix $M^d_{U,T}$ as well as his tags, if any (see step 3 of Figure 5.3). Once the matrix built, we proceed to the computation of the weights associated to each cell as detailed in the following.

5.2.2.2 Weighting the Users-Tags Matrix

As explained before, our approach relies on its ability to compute, for a given document d, an $m \times n$ Users-Tags matrix of m users and n tags where w_{ij} represents the extend to which the user u_i believes that the term t_j is associated to the document d.

Example 5.3. The tagging actions of *Alice* regarding the web page *Youtube.com* can be summarized as mixtures of two tags, *Info* and *Web*. Hence, we can suppose that the distribution of this two tags in this web page according to Alice is 50% for *Info* and 50% for *Web*. We refer to the distribution of a tag t_j in a document d according to a user u_i as: *the personal weight of* t_j *in* d *according to* u_i.

Hence, the main challenge here is *how to effectively estimate the personal weight of a tag* t_j *in a document* d *according to a user* u_i*?* We propose to use an adaptation of the well known *tf-idf* measure to estimate this weight. Hence, we define the weight w_{t_i} of the term t_i in a document d according to a user u_i as the *user term frequency, inverse document frequency (utf-idf)*, which is computed as follows:

$$w_{ij} = utf - idf = \frac{n^d_{u_i,t_j}}{\mid T_{u_i,d} \mid} \times log\left(\frac{|D_{u_i}| + 1}{|D_{u_i,t_i}|}\right) \tag{5.3}$$

where $n^d_{u_i,t_j}$ is the number of times u_i used t_j to annotate d computed after stemming. A high weight in *utf-idf* is reached by a high user term frequency and a low document frequency of the term in the whole set of documents tagged by the user; the weights hence tend to filter out terms commonly used by a user (see Step 4 of Figure 5.3).

At the end of this step, we obtain a matrix capturing the closest users (and their tags) to the query issuer, and this for each document that potentially match the query. Intuitively, the query issuer may have never annotated one of these documents, since the distribution of web pages over users follows a power law in folksonomies [HKGM08, SBC+10]. Given that, and due to the fact that a user is in average expected to use few terms to annotate a web page, we propose to infer a PSDR of this web page to that user based on other users feedback, translated by the inference of missing values in the Users-Tags matrix. This inference process is operated through matrix factorization, which is detailed in the next section.

5.2.3 Matrix Factorization

In the previous steps, we showed how we represent a document that matches a query using a Users-Tags matrix. This latter is expected to contain as relevant information as possible for the query issuer and the document by selecting relevant users and their tags as explained above. Each row i in the Users-Tags matrix of a given document represent the personal representation of the user u_i. However, this matrix is sparse, since it contains many missing values that should be inferred to compute the PSDR of the query issuer in particular. Hence, the problem at this point is to predict this missing values effectively and efficiently by employing other users feedback. One way to do so is to use a process of matrix factorization.

Matrix factorization has proven its effectiveness in both quality and scalability to predict missing values in sparse matrices [DFDF04, MKL09, MYLK08, MZL+11, NST+12, SM08]. This technique is based on the reuse of other users experience and feedback in order to predict missing values in a matrix. Concretely, to predict these missing values, the Users-Tags matrix is first factorized into two latent features matrices of users and tags. These latent features matrices are then used to make further missing values prediction. In its basic form, matrix factorization characterizes both users and tags by vectors of factors inferred from identifying weighting patterns. Therefore, the Users-Tags matrix $M^d_{U,T}$ of the web page *Youtube.com* is factorized using $M'^d_U \times M^d_T$, where the low-dimensional matrix M^d_U denotes the user latent features, and M^d_T represents the low-dimensional tag latent features.

Example 5.4. If we use 2 dimensions to factorize the matrix obtained above for weighting prediction (Step 4 of Figure 5.3), we obtain the matrices illustrated in the Step 5 of Figure 5.3. Note that $M^d_{u_i}$ and $M^d_{t_j}$ are the column vectors and denote the latent feature vectors of user u_i and tag t_j for the web page *Youtube.com*, respectively. Then we can predict missing values w_{ij} using $M'^d_{u_i} \times M^d_{t_j}$. Each row i of the predicted matrix

$M_U^{'d} \times M_T^d$ represents the personal representation of the i^{th} user according to this web page.

Notice that even if a user doesn't annotate a web page, this approach still can predict reasonable weightings as shown in Section 5.5.2. Also, it is important to mention that the solution of M_U^d and M_T^d is not unique (it depends on several parameters, e.g. the number of latent dimensions or the initial values of the factorization).

A matrix factorization seeks to approximate the Users-Tags matrix $M_{U,T}^d$ constructed above by a multiplication of l-rank factors, as follows:

$$M_{U,T}^d \approx M_U^{'d} \times M_T^d \tag{5.4}$$

where $M_U^d \in R^{l \times m}$ and $M_T^d \in R^{l \times n}$. Hence, we can approximate the Users-Tags matrix $M_{U,T}^d$ by minimizing the sum-of-squared-errors objective function over the observed entires as follows:

$$\arg \min_{M_U^d, M_T^d} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (M_{u_i,t_j}^d - M_{u_i}^{'d} \times M_{t_j}^d)^2 \tag{5.5}$$

where I_{ij} is the indicator function that is equal to 1 if user u_i used the tag t_j to annotate the document d_i and equal to 0 otherwise. In order to constrain the objective function in Equation 5.5 and to reduce the solution space, two regularization terms are added into Equation 5.5 [DFDF04, MYLK08, MKL09, SM08]. These two regularization terms ensure that the sum of the squared values of the two matrices $M_U^{'d}$ and M_T^d is as minimal as possible. Hence, the objective function becomes:

$$\begin{aligned} \arg \min_{M_U^d, M_T^d} \mathcal{L} = \arg \min_{M_U^d, M_T^d} & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (M_{u_i,t_j}^d - M_{u_i}^{'d} \times M_{t_j}^d)^2 \\ & + \frac{\lambda}{2} (\|M_U^d\|_F^2 + \|M_T^d\|_F^2) \end{aligned} \tag{5.6}$$

where $\lambda > 0$. Adding other regularization terms specific to our objective to constrain the objective function is part of our future work.

The optimization problem in Equation 5.6 minimizes the sum-of-squared-errors between observed and predicted weighting. A gradient based approaches, which is a first-order optimization algorithm can be easily applied to find a local minimum. It is based on the observation that if a function $f(x_1, \ldots, x_n)$ is defined and differentiable in a neighborhood of a point a, then $f(x_1, \ldots, x_n)$ decreases faster if one goes from a in the direction of the negative gradient of $f(x_1, \ldots, x_n)$, i.e. from a to $b = a - \mu \nabla f(x_1, \ldots, x_n)$. Hence, for $\mu > 0$ a small enough number, we have $f(a) \geq f(b)$.

Consequently, a local minimum of the objective function given by Equation 5.6 can be found by performing gradient descent in feature vectors $M_{u_i}^d$ and $M_{t_j}^d$, where we have:

$$\frac{\partial \mathcal{L}}{\partial M_{u_i}^d} = \sum_{j=1}^{n} I_{ij} (M_{u_i}^{'d} \times M_{t_j}^d - M_{u_i,t_j}^d) M_{t_j}^d + \lambda M_{u_i}^d \tag{5.7}$$

$$\frac{\partial \mathcal{L}}{\partial M^d_{t_j}} = \sum_{i=1}^{m} I_{ij}(M'^d_{u_i} \times M^d_{t_j} - M^d_{u_i,t_j})M^d_{u_i} + \lambda M^d_{t_j} \tag{5.8}$$

Algorithm 5.1 summarizes the gradient based algorithm for factorizing a Users-Tags matrix.

Algorithm 5.1 Gradient algorithm for matrix factorization.

Require: M^d_T: a Users-Tags matrix; l: a number of latent dimensions; μ: a step length; ϵ_1: a stop criteria.

1: Initialize M^d_U and M^d_T with a random Gaussian distributed double value $\mathcal{N}(0.0, 1.0)$.
2: **while** $||\nabla \mathcal{L}_k|| > \epsilon_1$ **do**
3: **for all** $u_{i,l}$ in $\mathcal{L}$ **do**
4: $u'^{k+1}_{i,l} = u^k_{i,l} - \mu \times \frac{\partial \mathcal{L}}{\partial M^k_{u_i}}$
5: **end for**
6: **for all** $t_{l,j}$ in $\mathcal{L}$ **do**
7: $t'^{k+1}_{l,j} = u^k_{l,j} - \mu \times \frac{\partial \mathcal{L}}{\partial M^k_{t_j}}$
8: **end for**
9: Compute $||\nabla \mathcal{L}_{k+1}|| = \mathcal{L}_{k+1} - \mathcal{L}_k$
10: Simultaneously update M^d_U and M^d_T
11: $k = k + 1$
12: **end while**

In summary, we should first initialize the matrices M^d_U and M^d_T. We choose to use the random Gaussian distributed double value $\mathcal{N}(0.0, 1.0)$ (line 1). Then, while the objective function given by Equation 5.6 has not converged, we update the value of each latent feature of each user and tag by going in the sens of the derivative with a step length μ (line 4 and 7 respectively).

Example 5.5. For factorizing the Users-Tags matrix of Figure 5.3 using two dimensions, we have the two multi-variables matrices:

$$M^d_U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} \end{bmatrix}, M^d_T = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} & t_{1,4} & t_{1,5} \\ t_{2,1} & t_{2,2} & t_{2,3} & t_{2,4} & t_{2,5} \end{bmatrix}$$

that we have to find. To do so, we have to minimize the following objective function:

$$\min_{M^d_U, M^d_T} \mathcal{L} = \min_{M^d_U, M^d_T} \frac{1}{2}[(0.5 - (u_{1,1}t_{1,1} + u_{2,1}t_{2,1}))^2$$

$$+(0.5 - (u_{1,2}t_{1,2} + u_{2,2}t_{2,2}))^2 + \ldots] + \frac{\lambda}{2}(\|M^d_U\|^2_F + \|M^d_T\|^2_F)$$

Having the derivative of the objective function for each of its variables, we can easily apply the gradient based Algorithm 5.1 to find M'^d_U and M^d_T that minimize it. The objective function will converge in eighteen iterations as illustrated in Table 5.2.

Table 5.2 – Values of the objective function of the matrix of Figure 5.3 in each iteration.

it 1	**it 2**	**it 3**	**it 4**	**it 5**	**it 6**	**it 7**	**it 8**	**it 9**
0.310	0.287	0.262	0.236	0.210	0.185	0.61	0.136	0.117
it 10	**it 11**	**it 12**	**it 13**	**it 14**	**it 15**	**it 16**	**it 17**	**it 18**
0.100	0.089	0.079	0.069	0.059	0.050	0.040	0.011	0.011

Once we have computed the factorized user latent features and tag latent features matrices, we can predict missing values using $M_U^{'d} \times M_T^d$. Then, we consider that:

Proposition 5.1. *The row that corresponds to the query issuer in the predicted matrix $M_U^{'d} \times M_T^d$ corresponds to his PSDR for the considered document. A PSDR is represented as a weighted vector of terms.*

Storing the PSDR of each document for each user is not suitable (this is like creating and storing an index structure for each user, which requires a lot of disk space). Therefore, we propose to execute this factorization process on the fly (at query time), since the complexity analysis performed in Section 5.2.5 shows that this approach scales linearly with the number of documents that match the query. In the next section, we describe our method to compute a ranking score for documents, w.r.t. their PSDR, their textual content, and the query.

5.2.4 Ranking Documents Using PSDR

In the previous sections, we showed how we can formalize a PSDR of a document that matches a query of a user. The PSDRs have to be matched to the query for quantifying their similarity while also considering the textual content of the document. Hence, we propose to compute a ranking score for documents using one of the following ranking functions:

1. A *Query Based Ranking Function* (QBRF), where the personalized ranking score of a document d that match a query q issued by a user u is computed as follows:

$$Rank_{QBR}(u, q, d) = \gamma \times Sim(\overrightarrow{q}, \overrightarrow{S_{d,u}}) + (1 - \gamma) \times SES(\overrightarrow{d}) \tag{5.9}$$

2. A *Profile Based Ranking Function* (PBRF), following the same idea as in [CZG+09, NM07, VCJ10, WJ10, XBF+08]. The personalized ranking score of a document d that matches a query q issued by a user u is computed as follows:

$$Rank_{PBR}(u, q, d) = \gamma \times Sim(\overrightarrow{p_u}, \overrightarrow{S_{d,u}}) + (1 - \gamma) \times SES(\overrightarrow{d}) \tag{5.10}$$

where, in both formulas, γ is a weight that satisfies $0 \leq \gamma \leq 1$, $SES(\overrightarrow{d})$ is the Search Engine Score (SES) given to the document d, e.g. we use the *Apache Lucene*

search engine in our implementation [32] [MHG10], $\overrightarrow{S_{d,u}}$ is the PSDR of the document d according to the user u, and $\vec{p_u}$ is the user profile constructed following Definition 2.8.

Inspired by the Vector Space Model (VSM), queries, documents, and PSDRs are modeled as vectors. Hence, we compute the similarities between these vectors using the cosine measure as follows:

$$Sim(\overrightarrow{q}, \overrightarrow{S_{d,u}}) = \frac{\overrightarrow{q} \bullet \overrightarrow{S_{d,u}}}{|\overrightarrow{q}| \times |\overrightarrow{S_{d,u}}|}, \; Sim(\overrightarrow{p_u}, \overrightarrow{S_{d,u}}) = \frac{\overrightarrow{p_u} \bullet \overrightarrow{S_{d,u}}}{|\overrightarrow{p_u}| \times |\overrightarrow{S_{d,u}}|} \tag{5.11}$$

At the end of this process we obtain a list of ranked documents according to (i) a matching between the textual content of documents and the query and (ii) the social interest of the user extracted from close relatives in the folksonomy. Finally, the top ranked documents are formatted for presentation to the user. In the next section, we propose a complexity analysis of our approach and we show the execution time needed to factorize a number of documents.

5.2.5 Complexity Analysis

The main computation effort for generating a PSDR of a document is in building the Users-Tags matrix and factorize it (Steps 1 to 5 in Figure 5.3). The time complexity needed for building a Users-Tags matrix is $O(\mid U_d \mid \times log(\mid U_d \mid))$, which corresponds to rank users for selecting the most representative (step 2 in Figure 5.3). For factorizing the matrix, the main computation of the gradient descent algorithm is evaluating the objective function $\mathcal{L}$ in Equation 5.6 and its derivatives in Equations 5.7 and 5.8 (see Algorithm 5.1). As pointed in [MYLK08], since the distribution of tags and users over documents in folksonomies follows a power law, the Users-Tags matrix is expected to be extremely sparse. Hence, the computational complexity of evaluating the objective function $\mathcal{L}$ is $O(\rho)$, where ρ is the number of nonzero entries in the Users-Tags matrix. Also, the computational complexity for the derivatives $\frac{\partial \mathcal{L}}{\partial M^d_{u_i}}$ and $\frac{\partial \mathcal{L}}{\partial M^d_{t_j}}$ of Equations 5.7 and 5.8 respectively are the same which is $O(\rho)$. Thus, the total computational complexity in one iteration of the gradient descent algorithm is $O(\rho)$. Consequently, for factorizing one document, the computational complexity is estimated to be $O(i \times \rho)$, where i is the number of iteration of the gradient algorithm (on average $i \backsimeq 15$ in our evaluations). Finally, for computing a PSDR of a given document, the time complexity is estimated to:

$$O(\mid U_d \mid \times log(\mid U_d \mid) + i \times \rho)$$

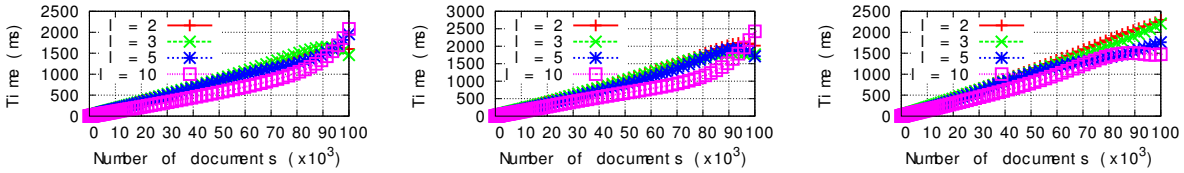
As a last step, the computational complexity for evaluating a query q that match m documents is estimated to:

32. http://lucene.apache.org/core/old_versioned_docs/versions/3_5_0/api/core/org/apache/lucene/search/Similarity.html

$$O(m \times [\mid U_d \mid \times log(\mid U_d \mid) + i \times \rho])$$

Since i, ρ and $\mid U_d \mid$ are estimated to a low values due to the sparse nature of folksonomies, we can say that the complexity scales linearly with the number of retrieved documents, which indicates that this approach can be applied to a very large datasets. By using parallel computation, we can easily and considerably reduce the execution time. This is part of our future work for improving the effectiveness of our approach.

As an illustration, Figure 5.4 shows the execution time needed for processing queries according to the number of documents that they match w.r.t. several parameters. These latter are: (i) l, the number of latent dimensions with which we perform the factorization, and (ii) k, the number of related users chosen to build the Users-Tags matrix. The queries and the users were randomly selected 10 times independently, and we report the average results each time. As depicted in Figure 5.4, none of these parameters have an impact on the execution time. This latter still scales linearly with the number of documents. Note that the average execution time of the factorization of a single Users-Tags matrix in our experiments was about 15μs. The factorization process was on average converging after 15 iterations. The results are obtained on a MacBook Pro with a 2.8GHz Intel Core i7 CPU and 4GB 1333MHz DDR3 of RAM, running MacOS X Lion v10.7.4.



(a) Top 1 related user. (b) Top 3 related users (c) Top 5 related users

Figure 5.4 – Execution time for processing queries with respect to the number of documents that they match.

5.3 Evaluation

In this section, we describe the evaluations we have performed on our approach. We have performed three types of evaluations:

1. A parameter estimation that aims to provide insights regarding the different values of the parameters used in our approach as well as their potential impact on the system.
2. A comparison study, where our approach is compared to other state-of-the-art approaches offline for providing insights about the obtained results and position the proposal.

3. An analysis of our approach to study its ability to achieve good performance even if users have annotated documents with few terms.

The evaluation has been performed on the *delicious* dataset described in Section 4.4.1 and Table 4.3. We follow the same evaluation methodology as the one described in Section 4.4.2. Note that our approach has been implemented using the *Apache Lucene* search engine.

5.4 Estimation of the Parameters

Our approach possesses several parameters that can be tuned and studied. While studying the impact of a parameter, we fix each time the others to the values described in Table 5.3.

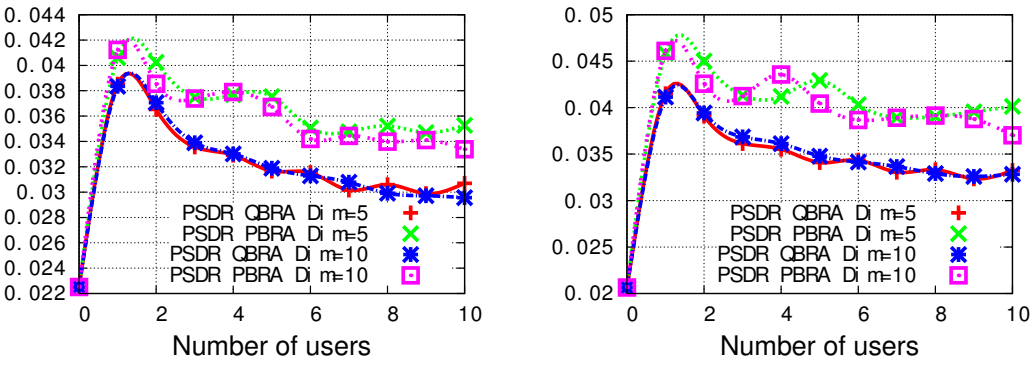
Table 5.3 – Default values of the parameters for their evaluation.

Parameter	Value	Remark
γ	1	To better estimate the impact of the PSDR on the other parameters
α	0	To better discriminate between users while varying the other parameters
Similarity	Cosine	/
Top users	2	/
Dimension	5 or 10	/
λ	0.02	/

Note that each time, we give the results obtained using: (i) two different dimensions for the factorization process (5 and 10), (ii) our two ranking functions, and (iii) our two retrieval processes.

5.4.1 *Impact of the Number of Users (k)*

This parameter is illustrated in Figure 5.5. The obtained results show that optimal results are obtained while selecting 1 or 2 related users depending on the ranking function and the retrieval process used. Adding more users decreases significantly the performance. This is due to the fact that the filtered out users have inappropriately annotated documents and are socially far from the query issuer. These users represent the irrelevant users that we would like to set aside. Thus, these results show the effectiveness of our ranking function proposed in Section 5.2.2.1.

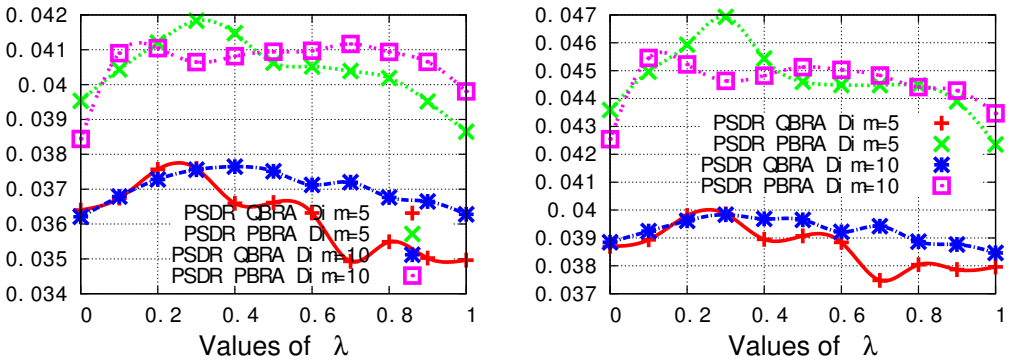


(a) Mean Average Precision. (b) Mean Reciprocal Rank.

Figure 5.5 – Impact of the number of users.

5.4.2 *Impact of the Regularization Terms (λ)*

The results of this parameter are illustrated in Figure 5.6. This parameter controls the weight of the social regularization terms of the objective function given in Equation 5.6. We set its value such as $0 \leq \lambda \leq 1$. We assume that we cannot assign to it a value higher than 1, to prevent neglecting the core of the objective function, which is the sum-of-squared-errors between the observed weighting of the Users-Tags matrix $M^d_{U,T}$ and the approximated weighting using $M'^d_U \times M^d_T$.



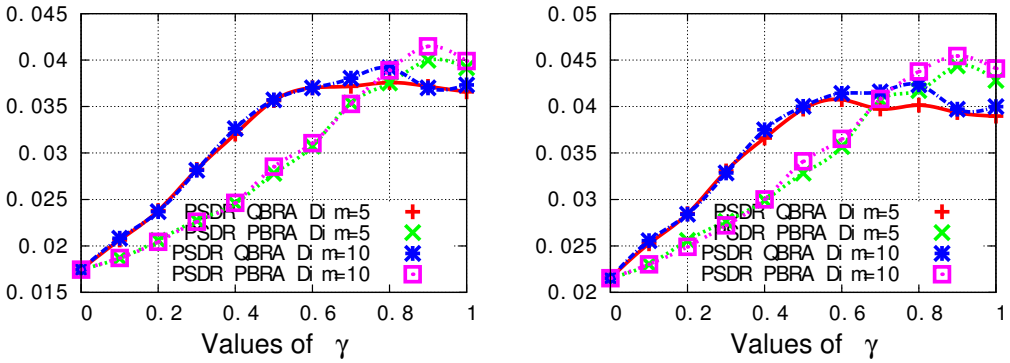
(a) Mean Average Precision. (b) Mean Reciprocal Rank.

Figure 5.6 – Impact of the regularization terms λ.

Even if the improvement when varying λ is not significant, it has an impact on the performance. Our method achieves the best performance when $\lambda \in [0.4, 0.8]$ depending on the number of latent dimensions with which we perform the factorization. This seems to be encouraging to further propose other regularization terms to constrain the objective function. This is part of our future work.

5.4.3 *Impact of the PSDR Score (γ)*

The results of this parameter are illustrated in Figure 5.7. The optimal value is obtained for $\gamma \in [0.6, 0.9]$, a value which we consider as a tradeoff between the personalized and the non-personalized parts.



(a) Mean Average Precision.

(b) Mean Reciprocal Rank.

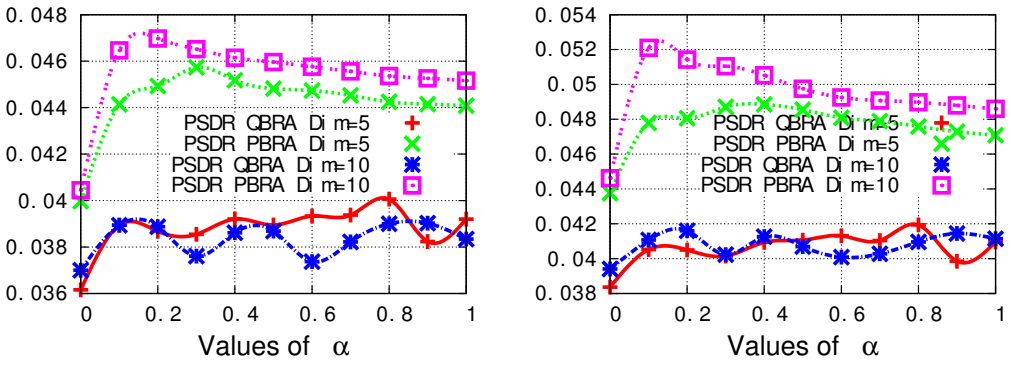
Figure 5.7 – Impact of γ.

5.4.4 *Impact of the social proximity part (α)*

The results of this parameter are illustrated in Figure 5.8. This parameter allows to control the social and the document proximity parts while computing the ranking scores for users in Equation 5.2. The obtained results show that optimal performances are obtained for $\alpha \in [0.1, 0.4]$, improving the MAP and MRR by $3 - 4\%$ for the QBRF and PBRF ranking functions. On the one hand, considering only the social proximity part doesn't provide good performance ($\alpha = 0$). This is due to the fact that there are a lot of users who have annotated relevant documents with relevant tags, and who do not share affinity with the query issuer. On the other hand, considering only the document proximity part doesn't necessarily provide a good retrieval performance ($\alpha = 1$). This is due to the fact that we are not taking into account the social dimension for discriminating between users.

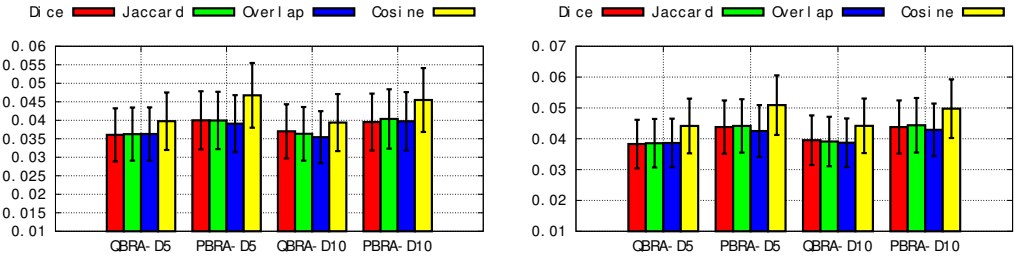
5.4.5 *Impact of the Similarity Measure*

The results of this parameter are illustrated in Figure 5.9. As one can see, the cosine similarity measure provides better retrieval performance by allowing to be more efficient in discriminating between users. This is certainly due to the fact that the cosine measure takes into account the importance of each tag for each user while computing similarities. The other similarities are purely statistical since they consider only the number of tags (in common) without estimating the importance of each of these tags.



(a) Mean Average Precision. (b) Mean Reciprocal Rank.

Figure 5.8 – Impact of α.



(a) Mean Average Precision. (b) Mean Reciprocal Rank.

Figure 5.9 – Impact of the similarity measure. 95% confidence intervals are shown.

5.5 Comparison with Baselines

Our objective here is to estimate how well our approach meets the users' information needs compared with other state-of-the-art approaches, objectively. Our approach is evaluated using the optimal values computed in the previous section while using five dimensions in the factorization process and our two ranking functions as explained in Section 5.2.4. We compare our approach to several personalized and non-personalized baselines, in which the social based score is merged with the textual based matching score using a linear function with a γ parameter. These baselines are summarized and described in Table 5.4.

5.5.1 Analysis of the Comparison

The results of the comparison are illustrated in Figure 5.10, while varying γ. In the following, we compare the retrieval performance of our approach with several personalized and non-personalized baselines.

Table 5.4 – Summary of the baselines.

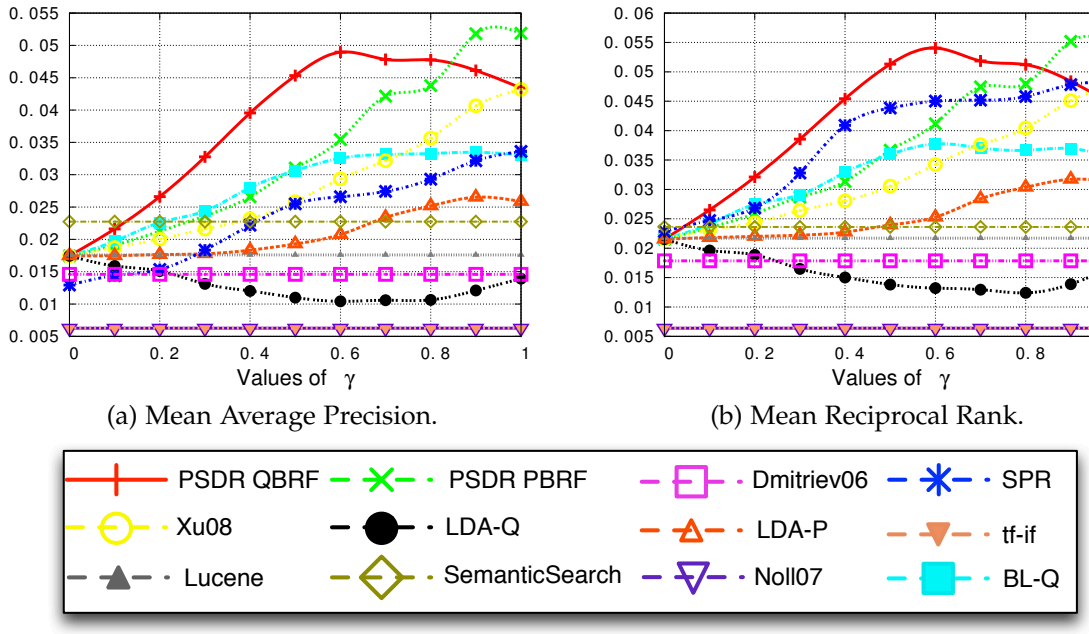
		Baseline	Description
Non-personalized approaches	1	**SPR [BXW+07]**	See Section 3.3.2.
	2	**Dmitriev06 [DEFS06]**	
	3	***B*L-Q**	This approach use a query based ranking function as described in Equation 5.9. However, we use a social representation of documents based on all their annotations weighted using the *tf-idf* measure.
	4	**Lucene**	This approach is the Lucene naive function, where all the parameters have been set to their default values [MHG10].
	5	**LDA-Q**	This approach use LDA [BNJ03] for modeling queries and documents. Then, for each document that matches a query, we compute a similarity between its topic and the topic of the query using the cosine measure (inferred using the previous constructed model). The obtained value is merged with the textual ranking score as in Equation 5.9.
Personalized approaches	6	**Xu08 [XBF+08]**	See Section 6.3.1.
	7	**Noll07 [NM07]**	See Section 6.3.3.
	8	**tf-if [VCJ10]**	See Section 6.3.4.
	9	**Semantic Search [BCK+08]**	See Section 6.3.5.
	10	**LDA-P**	See Section 6.3.2.

5.5.1.1 PSDR vs Non-Personalized Approaches

As illustrated in Figure 5.10, the obtained results show that our approach is much more efficient than all the non-personalized approaches for all values of γ. Hence, we conclude that the personalization efforts introduced by our approach in the representation of documents with respect to each user bring a considerable improvement of the search quality. We also notice that most of the non-personalized approaches decease their performance for high values of γ. This is certainly due to the fact that they are not designed for personalized search, since these approaches fail in discriminating between users in spite of their preferences.

5.5.1.2 PSDR vs Personalized Approaches

Here, the obtained results also show that our approach is much more efficient than all the personalized approaches for all values of γ (except for $\gamma = 0$, where Semantic Search gives better results). Especially, our approach outperform the LDA-P approach



(a) Mean Average Precision.

(b) Mean Reciprocal Rank.

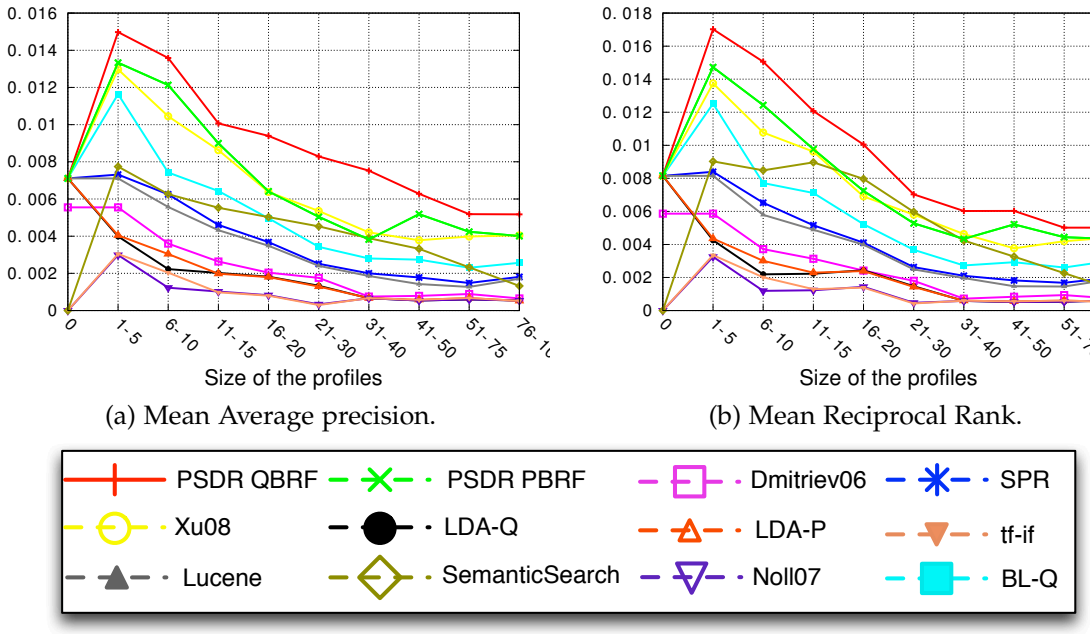
Figure 5.10 – Comparison with the baseline while varying γ and using the optimal values of the parameters.

and the Xu08 approach, which we consider as the closest works to our. We also notice that the Noll07 and the tf-if approaches give poor results. This is certainly due to the fact that they fail in ranking documents that doesn't share tags with users, since in our experiment we remove the triplets that associate the user, the query terms and documents.

5.5.2 Performance on Different Queries

In this section, we study the ability of our approach to achieve good performance even if users have annotated documents with few terms. Hence, to do so, we propose to compare our approach with the other baselines while following the same evaluation process as described in Section 4.4.2. We select 2000 query pairs (u, t) based on the number of tags the users used in their tagging actions. The query pairs are grouped into 10 classes: "0", "1-5", "6-10", "11-15", "16-20", "21-30", "13-40", "41-50", "51-75", and "76-100", denoting how many tags users have used in their tagging actions, e.g. class "1-5" is composed with users who have a profile length between 1 and 5. Note that we select the optimal values of the parameters of the PDSV framework as discussed in Section 5.4, while fixing $\gamma = 0.5$ for all the approaches.

The experimental results are shown in Figure 5.11 over the 10 classes of queries. The obtained results show that the PSDR approach outperforms almost all the baseline approaches for all the queries. We also report that even if a user doesn't annotate a web page, the PSDR approach still can improve the search quality comparing



(a) Mean Average precision.

(b) Mean Reciprocal Rank.

Figure 5.11 – Performance comparison on different queries, while fixing $\gamma = 0.5$.

to other approaches. This is certainly due to the fact that reasonable weighting are predicted in the Users-Tags matrix since the explicit feedback of the closest users is used to compute a PSDR of each document that potentially match the query. These results show the effectiveness of the PSDR approach in the context of sparse data.

The results of this offline evaluation show that our approach is much more efficient than all the baselines even if the query issuer doesn't annotate a web page. Especially, our approach outperforms all the personalized approaches, which we consider as the closest works to our. Hence, we conclude that the personalization efforts introduced by our approach in the representation of documents with respect to each user bring a considerable improvement for the search quality.

Finally, we note that in this offline evaluation, the better performances are obtained while using QBRF and choosing one or two of the most related users to the query issuer. However, these results should be reinforced using an online evaluation to give a better overview of the performance, which is performed through a user survey. This part is detailed in the next section.

5.6 User Survey

For the user study, we used our *delicious* dataset from which we selected 335 pairs of queries and users. These users are considered as query initiators and have used all the selected query tags at least once on the same document. We then ran the queries using our approach and the baselines that performed the best in the offline evaluation,

i.e. At each iteration, the user is presented with lists of ranked documents using (i) our approach and (ii) a randomly selected baseline algorithm. Each list contains seven documents. Note that for all approaches, γ was set to 0.5.

In the assessment phase, 39 volunteers participated to judge the relevance of the results. Each volunteer (who is considered as a query initiator) was shown, in addition to the results for the query from the pool: (i) the documents from the query initiator that contain at least one of the query tags and (ii) the tags he used in his tagging actions. This is to help the volunteers to understand the personal context of the (real) query initiators as well as their interests. This way, we intend to overcome the aforementioned problem of subjectively assessing result qualities with the eyes of the query initiator.

Once a list is presented to a participant, he/she marks each result as: very relevant, relevant, or irrelevant w.r.t the context of the (real) query initiator. This process is performed by the evaluator without knowing which algorithm generated the result. Figure 5.12 shows the interface that the users obtained when they participated to the survey. This interface contains (i) the tags the user used in his tagging action (in the top right part), (ii) the documents he tags with the query terms (in the right part), and (iii) the two lists of results to be judged after the query was issued.

Figure 5.12 – User survey web page.

The quality of each result was measured by the normalized discount cumulative gain (nDCG@7) and by precision at 10 (P@7), averaged over the set of judged queries. For DCG calculation, we used gains (2,1,0) for the three relevance levels respectively, and the discount function used was is given in Equation 2.6. Normalization (nDCG) was done by dividing the DCG value with an ideal DCG value calculated as all results are highly relevant. For P@7 calculation, we considered any positive judgment as relevant. The obtained results are shown in Figure 5.13 as measured by NDCG@7 and P@7.

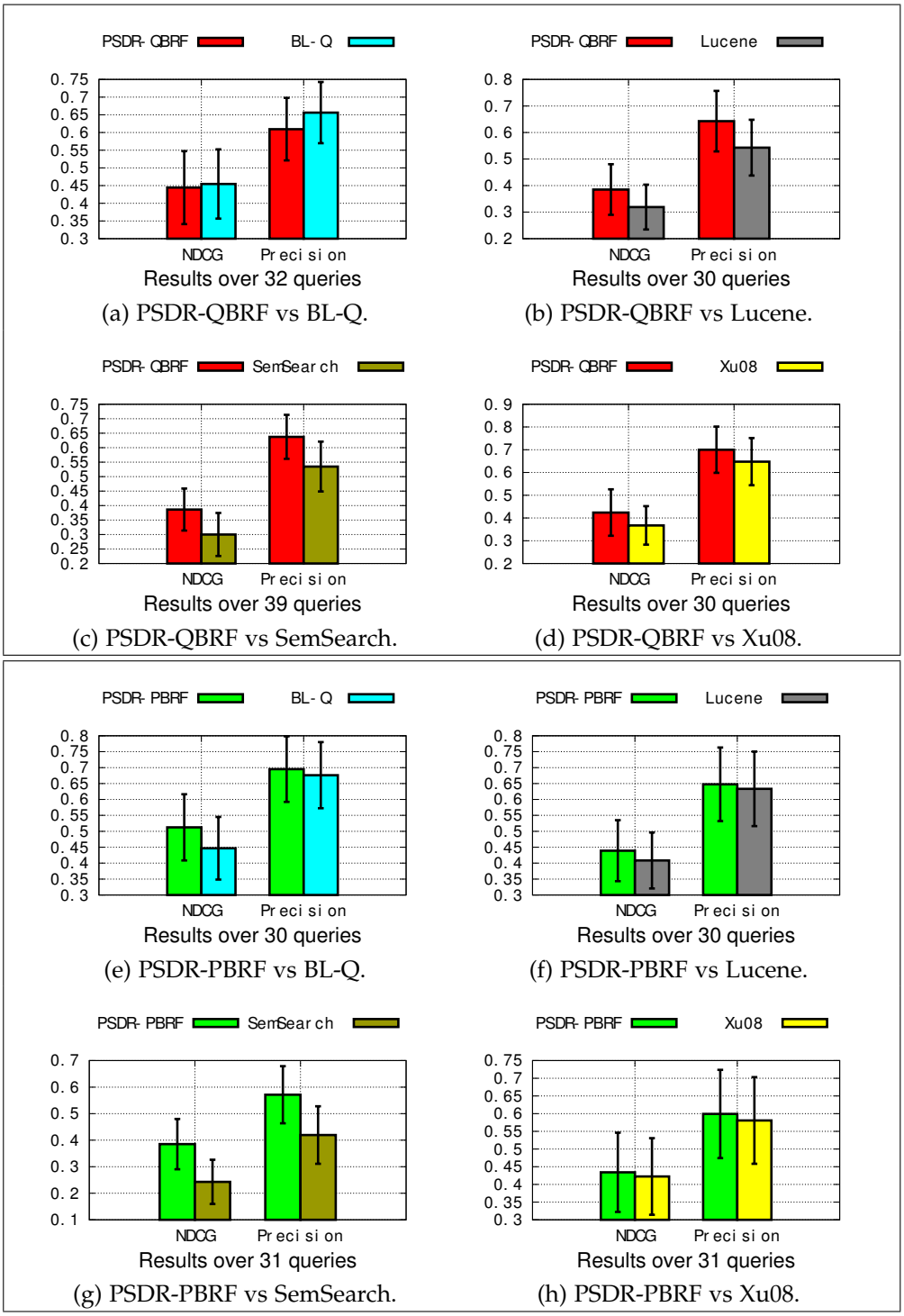
The main outcome of the survey can be summarized as follows: (i) this user survey confirms, to some extent, the results obtained in the offline evaluation since the PSDR approaches outperform the selected baselines. (ii) The BL-Q approach, even if it is a non-personalized approach, has been judged to be more efficient than the PSDR-QBRF approach. (iii) The advantage observed by the PSDR approaches is not as important. Actually, several participants mentioned the difficulty in judging the relevance of the queries, mostly because of unfamiliarity with the users they are related to. (iv) We believe that the best performance is provided by the PSDR-PBRF approach since it outperforms the baselines. This remark should be confirmed by evaluating the two PSDR approaches together on the same queries as this has been done with the baselines.

As a conclusion for this evaluation, we notice that there are several substantial differences between the two evaluation methods. Both methods confirm the significant contribution of the personalization introduced in the representation of documents using the PSDR approach, and the superiority of using it for ranking purposes. However, the results obtained in the offline evaluation show the superiority of the PSDR-QBRF approach overs the PSDR-PBRF approach, which is not what we observe in the user survey. Also, although the superiority of the PSDR approach has been observed in the offline evaluation, the user survey showed some subtlety regarding this superiority, i.e. in the user survey, the superiority of the PSDR approach is not so obvious.

5.7 Conclusion and Future Work

This chapter discusses a contribution to the area of IR modeling while leveraging the social dimension of the web. We proposed a Personalized Social Document Representation framework (PSDR), an attempt to use social information to enhance and provide a personalized representation of documents to users. While a user submit a query, we construct on the fly a PDSR of all documents that potentially match the query based on other users' experience (while considering both users that are social close to the query issuer and relevant to documents). Then, we rank these documents with respect to one of the two ranking functions that we propose. The complexity analysis shows that improving the IR process at this stage is possible with relatively an acceptable execution time. Also, the experiments that we have performed on a *delicious* dataset show the benefit of such an approach compared to others.

Even with the interest of the proposed method, there are still possible improvements that we can bring. We are investigating the possibility of using parallel computation to reduce the execution time. We are also investigating ways to add social regularization terms to the objective function of the matrix factorization in order de constrain it and reduce the solution space of factorization. The temporal dimension of social users' behavior has not been investigated yet. This is also part of our future work to improve our proposal.



(a) PSDR-QBRF vs BL-Q. (b) PSDR-QBRF vs Lucene.

(c) PSDR-QBRF vs SemSearch. (d) PSDR-QBRF vs Xu08.

(e) PSDR-PBRF vs BL-Q. (f) PSDR-PBRF vs Lucene.

(g) PSDR-PBRF vs SemSearch. (h) PSDR-PBRF vs Xu08.

Figure 5.13 – Results of the PSDR user survey: The precision of the search results for different algorithms measured by nDCG@7 and P@7.

Chapter 6

Ranking Functions For Personalized Web Search Using Folksonomy

In this chapter, we first present a state of the art on personalized social ranking functions. Then, based on some technical weaknesses of these ranking functions, we present SoPRa, a new social ranking function, which considers the social dimension of the Web. This social dimension can be any social information that surrounds documents along with the social context of users. SoPRa has been evaluated through an offline study and through a user survey over a large public dataset of *delicious*, and compared to the closest state of the art methods. The obtained results show significant benefits for personalized search.

6.1 Introduction

In IR systems, a relatively large number of Web pages can match users' queries. Therefore, effectively ranking these Web pages is a key problem, since a user cannot browse all these Web pages. Ranking in IR is part of the modeling process (which also includes the definition of a conceptual model for representing documents and queries). It consists of the definition of a ranking function that allows quantifying the similarities among documents and queries. A ranking function should incorporate many features to be effective, e.g. features of the document, the query, the overall document collection, the user, etc. Many of these features are summarized in Table 6.1. In this chapter, we focus on personalized social ranking functions. These ranking functions are expected to provide personalized search results, while leveraging the social context that surrounds both users and documents.

While relying on the Vector Space Model (VSM, Section 2.2.2), on the one hand, we propose to review many personalized social ranking functions that improve the ranking process by personalizing it using social features, i.e. social annotations. Here, we intend mainly to answer the following questions: *What are these functions and how do they work? What is the context where each function is more efficient? What is the best*

Table 6.1 – Features for defining a ranking function.

Features of users	**Features of documents**	**Features of queries**
- Query logs - Profiles - Annotations - Comments - Tweets - Ratings - etc...	- Textual content, e.g. any word that appears in a document - Metadata, e.g. description, keywords, title, etc... - Annotations - Queries - Anchor-text - etc...	- Length of the query, i.e. number of terms - Frequency of terms - etc...

ranking function? To which extent are they efficient compared to non-personalized approaches? can we propose a new function to cope with their limitations?

On the other hand, based on the technical issues identified in the state of the art of the personalized social ranking functions, we propose a new ranking function that leverages the social dimension of the Web, e.g. leveraging the social context of Web pages and users. Currently, our approach considers social metadata related to documents and users on social bookmarking systems. However, it can easily be extended to consider other social metadata, e.g. comments, tweets, etc.

The main contributions of this chapter can be summarized as follows:

1. A **So**cial **P**ersonalized **Ra**nking function called SoPRa. SoPRa leverages both the social context of users and documents for ranking purposes;
2. An extension of SoPRa that considers entities individually;
3. A deep study of the state of the art of ranking functions in social collaborative setting;
4. A deep analysis of the performances of these personalized social ranking functions and a comparison with non-personalized social approaches. We show to which extent SoPRa contributes to an efficient Web search at the expense of existing approaches;
5. An end-user evaluation for the results' quality of SoPRa;
6. Finally, a discussion on the effectiveness, the weakness and the performance of each approach in different contexts.

This chapter is organized as follows: in Section 6.2, we formally define the ranking problem we tackle. In Section 6.3, we review many of the personalized ranking functions studied in this paper. Section 6.4 introduces SoPRa, our approach for ranking documents. The evaluations are presented and discussed in Sections 6.5 and 6.6. Finally, we conclude and provide some future directions in Section 6.7.

6.2 Problem Definition of Ranking

Within the context of folksonomies, we can formalize the ranking problem as follows: Let's consider a folksonomy $\mathbb{F}(U, T, R)$ from which a user $u \in U$ submits a query q to a search engine. We would like to re-rank the set of resources $R_q \subseteq R$ (or documents) that match q, such that relevant resources for u are highlighted and pushed to the top for maximizing his satisfaction and personalizing the search results.

The ranking should follow an ordering $\tau = [r_1 \geq r_2 \geq \cdots \geq r_k]$ in which $r_k \in R$ and the ordering relation is defined by $r_i \geq r_j \Leftrightarrow Rank(r_i, u, q) \geq Rank(r_j, u, q)$, where $Rank(r, u, q)$ is a ranking function that quantify similarity between the query and the resource w.r.t the user [VCJ10].

6.3 Personalized Ranking Functions Based on Folksonomies

In this section, we formally define the different personalized ranking functions studied in this chapter. We each time present the ranking score of a document d for a query q issued by a user u noted $Rank(d, q, u)$.

6.3.1 Profile Based Personalization

The approach presented by Xu et al. [XBF+08] assumes the ranking score of a document d in the result list when a user u issues a query q is decided by two aspects: (i) a textual matching between q and d, and (ii) a user interest matching between u and d. Hence, following our notation in Table 2.1, their approach can be defined as follows:

$$Rank(d, q, u) = \gamma \times Cos(\overrightarrow{p_u}, \overrightarrow{T_d}) + (1 - \gamma) \times Cos(\overrightarrow{q}, \overrightarrow{d}) \tag{6.1}$$

where, γ is the weight that satisfies $0 \leq \gamma \leq 1$, $Cos(\overrightarrow{q}, \overrightarrow{d})$ denotes the textual matching score between d and q, and $\overrightarrow{T_d}$ is the vector that models the social representation of the document d. Currently, the weighting scheme used is the *tf-idf* [BYRN11]. We refer to this approach as Xu08.

6.3.2 Topics Based Personalization

We present here a topics-based approach, to which we refer as LDA-P. This approach is based on Latent Dirichlet Allocation (LDA) [BNJ03]. LDA is an unsupervised topic modeling approach, where each topic consists of a group of words. Thus, documents are viewed as a composition of probabilistic topics that are represented as a T dimensional random variable θ. For each document, the topic distribution θ has a Dirichlet prior $p(\theta|\alpha) \sim Dir(\alpha)$. As in generative models, LDA generates each document by first picking a topic distribution θ from the Dirichlet prior and then use

each document's topic distribution to sample latent topic variables z_i. LDA makes the assumption that each word is generated from one topic, where z_i is a latent variable indicating the hidden topic assignment for word w_i. Probability of choosing a word w_i under topic z_i , $p(w_i|z_i, \beta)$, depends on different documents.

LDA-P relies on the fact that the set of tags can be used to represent Web pages and as input for LDA to construct a model. Then, for each document that matches a query, LDA-P computes a similarity between its topic and the topic of the user profile using the cosine measure (inferred using the previous constructed LDA model). The obtained similarity value is merged with the textual ranking score to provide a final ranking score for a document that matches a query w.r.t the query issuer as follows:

$$Rank(d, q, u) = \gamma \times Cos(\overrightarrow{u_{topic}}, \overrightarrow{d_{topic}}) + (1 - \gamma) \times Cos(\overrightarrow{q}, \overrightarrow{d}) \tag{6.2}$$

where, $\overrightarrow{u_{topic}}$ and $\overrightarrow{d_{topic}}$ are respectively the vectors that model the user and the document topics based on the constructed LDA model. Note that we use a Java implementation of LDA using Gibbs Sampling for Parameter Estimation and Inference [33]. In each execution, we use the default values proposed by this implementation, i.e. $\alpha = 0.5$, $\beta = 0.1$, $topics = 100$, and a number of most likely words for each topic equal to 20.

6.3.3 Scalar Tag Frequency Based Personalization

The approach presented by Noll and Meinel [NM07] considers only a user interest matching between u and d. This approach does not make use of the user and document length normalization factors, and only uses the user tag frequency values. The authors normalize all document tag frequencies to 1, since they want to give more importance to the user profile when computing the similarity measures. Following the notation given in Table 2.1, their ranking function can be defined as follows:

$$Rank(d, q, u) = \sum_{t \in T_u \wedge t \in T_d} |D_{u,t}| \tag{6.3}$$

We refer to this approach as Noll07.

6.3.4 Scalar tf-if Based Personalization

Vallet et al. [VCJ10] proposed to improve the Noll07 approach above by including a weighting scheme based on an adaptation of the *tf-idf* called *term frequency-inverse frequency* (*tf-if*). Thus, the document tag frequencies are no more normalized to 1 as in Noll07. Rather, both document tags and user tags frequencies are replaced with their tf-if measure in order to perform a multiplication between the two vectors. This ranking function is defined as follows:

33. http://jgibblda.sourceforge.net/

$$Rank(d,q,u) = \sum_{t \in T_u \wedge t \in T_d} (tf_u(t) \times iuf(t) \times tf_d(t) \times idf(t)) \tag{6.4}$$

where $tf_u(t)$ is the frequency of use of t by u, $iuf(t)$ is the user term frequency, $tf_d(t)$ is the frequency of occurrence of t in d, and $idf(t)$ is the inverse document frequency. We refer to this approach as tf-if.

6.3.5 Affinity Based Personalization

Bender et al. [BCK+08] proposed several personalized ranking functions based on relations in a folksonomy. More precisely, we describe in this paper the following two ranking functions that we consider as relevant to our study:

1. Semantic Search: This approach ranks documents by considering users that hold similar content to the query, i.e., users who used at least one of the query terms in describing their content. This ranking function is defined as follows:

$$Rank(d,q,u) = \sum_{t \in q} \sum_{u_i \in U_t} \frac{k_1 + tf_u(t)}{K + tf_u(t)} \times log\left(\frac{|D_u| - |D_{u,t}| + 0.2}{|D_{u,t}| + 0.5}\right) \tag{6.5}$$

   where K is a value that describe the average length of the document, and k_1 is a constant set to 1.2. We refer to this approach as SemanticSearch.

2. Social Search: This approach ranks documents by considering friends of the query issuer who used at least one of the query terms in describing their content.This ranking function is defined as follows:

$$Rank(d,q,u) = \sum_{t \in q} \sum_{u_i \in F(u)} cos(\overrightarrow{p_u}, \overrightarrow{p_{u_i}}) \times \frac{k_1 + tf_u(t)}{K + tf_u(t)} \times log\left(\frac{|D_u| - |D_{u,t}| + 0.2}{|D_{u,t}| + 0.5}\right) \tag{6.6}$$

   where $F(u)$ is the set of users who are similar to the query issuer, i.e. users who used similar tags to those of the query issuer. We refer to this approach as SocialSearch.

The six personalized ranking functions described are all based on the social context that surrounds users and documents in a folksonomy. However, these ranking functions have some technical issues, among which:

(i) As illustrated in Figure 6.1, these ranking functions consider, either a textual matching between the query and the document, or a user interest matching between the user profile and the document. However, we can imagine other possibilities such as, a matching between the query and the social representation of a document, a matching between the user profile and the textual content of a document, etc.

(ii) These ranking functions consider the social representation of documents without discriminating between contributors, i.e. the different users who annotate documents. We believe that we should process each user, who annotates a Web page individually, in order to fully leverage the collaborative setting, i.e. similarity and proximity between users, trustworthiness and confidence of users, etc.

Based on these two technical issues, we propose a new ranking function called SoPRa, which is detailed in the next section.

6.4 SoPRa Function

In the following, we first define the SoPRa ranking function, and then we present the methods for modeling social documents and users. This basic version of SoPRa is expected to address problem (i). Finally, we present an extended version of SoPRa, which is expected to tackle both problem (i) and (ii).

6.4.1 Basic SoPRa

On the one hand, we believe that a matching score between a document d and a query q should be based on (i) a textual matching score, and (ii) a social matching score. The textual matching score expresses the similarity between the textual content of d and q. The social matching score expresses how similar the social representation of d is, for q. This social representation is based on the annotations associated to d. More formally, in this work, we consider these two ranking scores as independent evidence, and we propose to merge them using a linear function as follows:

$$Score(q,d) = \beta \times Cos(\overrightarrow{q}, \overrightarrow{T_d}) + (1-\beta) \times Cos(\overrightarrow{q}, \overrightarrow{d}) \tag{6.7}$$

where, $Cos(\vec{A}, \vec{B})$ is the cosine similarity measure computed using Equation 2.3, β is a weight that is equal to 0.5, $Cos(\overrightarrow{q}, \overrightarrow{d})$ is currently computed using the *Apache Lucene* search engine in our implementation, and $\overrightarrow{T_d}$ is the vector that models the social representation of the document d.

On the other hand, in the non-personalized search engines (classic IR models), the relevance between a query and a document is assumed to be only based on the textual content of the document. However, as relevance is actually relative for each user, considering only a matching between a query and documents is not enough to generate satisfactory search results. Thus, we propose to estimate the interest of a user u to a document d by computing a similarity between the profile of u and the social representation of d. Then, we propose to merge this interest value to the previous ranking score computed in Equation 6.7 for computing an overall score to a document. Formally, the ranking score of a document d that potentially match the query q issued by a user u is computed as follows:

$$Rank(d, q, u) = \gamma \times Cos(\overrightarrow{p_u}, \overrightarrow{T_d}) + (1 - \gamma) \times Cos(q, d) \tag{6.8}$$

where, γ is the weight that satisfies $0 \leq \gamma \leq 1$. The ranking model of the basic SoPRa function is illustrated in Figure 6.1 for more clarity.
In summary, SoPRa ranks documents according to: (i) a textual content matching score of documents and the query, (ii) a social matching score of documents and the query, and (iii) the social interest score of the user to documents.

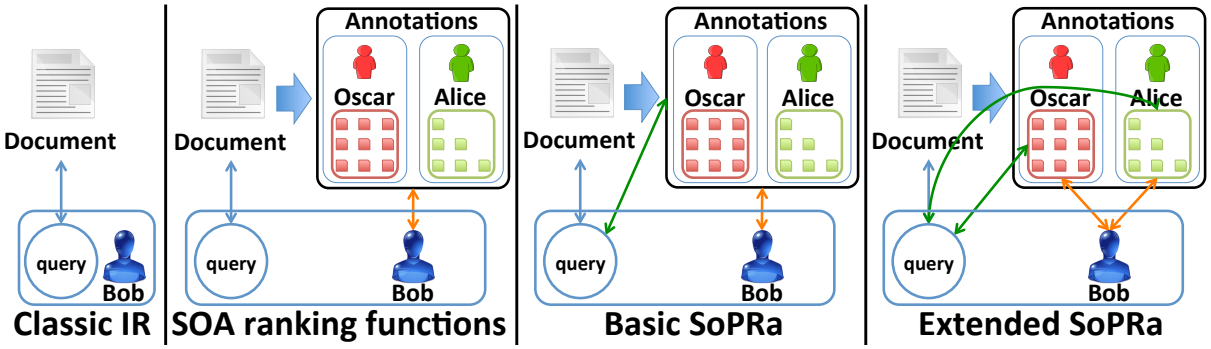


Figure 6.1 – Illustration of the basic differences between the different approaches.

6.4.2 Weighting scheme

In this chapter, the social representations of documents and the user profiles are estimated by their social annotations and modeled as in the VSM. Hence, if we consider Web pages or users as documents and annotations as terms, the above setting is right for the VSM. Even if the VSM has been developed a long time ago, it has shown its effectiveness for IR and remains very competitive and challenging. One of the key points in the VSM is the weighting of terms. We propose to simply weight annotations using the *tf-idf* measure as follows:

$$w_t^d = tf_t \times log(\frac{\mid R \mid}{\mid R_t \mid}),\ w_t^u = utf_t \times log\left(\frac{\mid U \mid}{\mid U_t \mid}\right) \tag{6.9}$$

where w_t^d is the weight of the term t in the social representation of d, tf_t denotes the tag frequency, w_t^u is the weight of the term t in the profile of u, and utf_u is the user term frequency, i.e. the number of time the user u used the tag t.

6.4.3 Extended SoPRa

In classic models of IR, the content of a Web page is considered as a mixture of homogeneous terms generated by the same creator, i.e. the author of the Web page. However, social bookmarking systems allow users to freely assign annotations to documents following their own vocabulary to describe these documents. Hence,

unlike the textual content of a Web page, annotations can be seen as a mixture of heterogeneous fragments, where each fragment describes the content of the Web page with annotations of a particular user. This notion of fragments is illustrated in Figure 6.1 as clusters of annotations. Consequently, we believe that IR ranking functions may be improved by considering independently each user that annotates a Web page. Strengthening annotations provided by similar users to the query issuer can enhance the score of a document. To address this problem, we propose an extension of SoPRa by discriminating between users who annotate Web pages and by considering their similarities with the query issuer. Hence, we extend the basic SoPRa as follows:

$$Rank(d,q,u) = \gamma \times \sum_{u_k \in U_d} Cos(\overrightarrow{p_{u_k}}, \overrightarrow{p_u}) \times Cos(\overrightarrow{p_u}, \overrightarrow{T_{u_k,d}}) + (1-\gamma) \times \left[\beta \times \sum_{u_k \in U_d} Cos(\overrightarrow{p_{u_k}}, \overrightarrow{p_u}) \times Cos(\overrightarrow{q}, \overrightarrow{T_{u_k,d}}) + (1-\beta) \times Cos(\overrightarrow{q}, \overrightarrow{d}) \right] \tag{6.10}$$

where $\overrightarrow{T_{u_k,d}}$ is the vector that models the social representation of the document d based only on the annotations provided by u_k to d. The ranking model of the extended SoPRa is illustrated in Figure 6.1 for more clarity.

In summary, in this section, we presented SoPRa in its basic form as well as an extension of SoPRa, which individually consider users and their similarities to the query issuer. In the next sections, we describe the evaluations we have performed on SoPRa and the ranking functions described previously.

6.5 Evaluation

In this section, we conduct several experiments to compare the retrieval quality of SoPRa and the personalized ranking functions described in Sections 6.4 and 6.3 respectively. We also compare these ranking functions to non-personalized ranking methods. Our experiments intend to address the following questions:

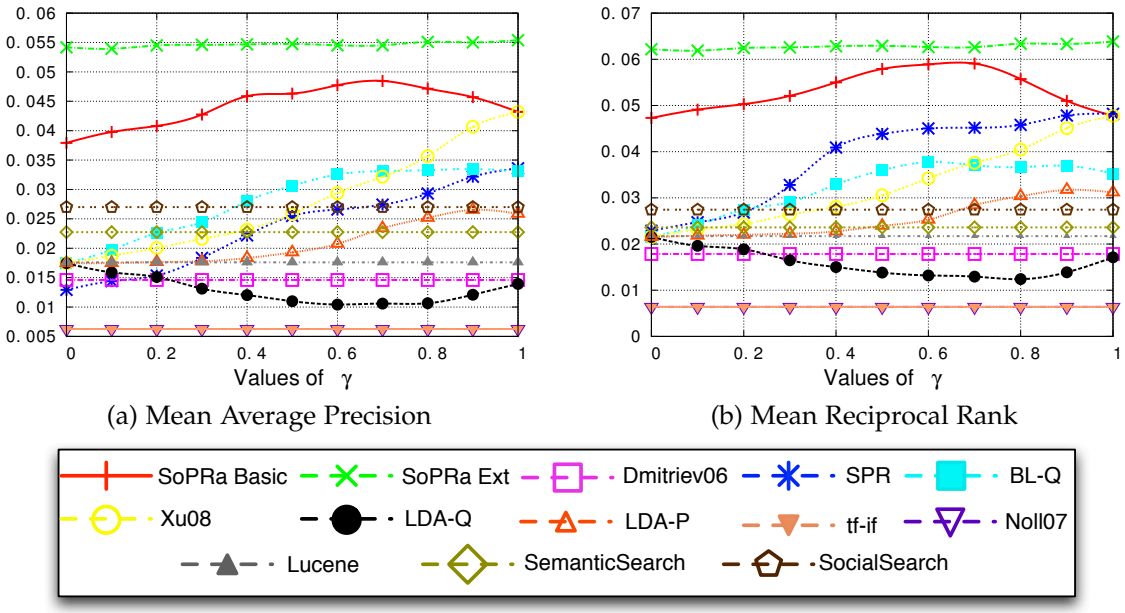
1. What is the effectiveness of these personalized ranking functions compared to the non-personalized ones?
2. What is the performance comparison on users with different profile configuration, i.e. length?
3. Can these personalized ranking functions achieve good performance even if users have no bookmarks?
4. What is the results' quality of SoPRa from an end-user perspective?
5. Are these personalized ranking functions efficient for large datasets?
6. What is the best personalized ranking function?

The evaluation has been performed on the *delicious* dataset described in Section 4.4.1 and Table 4.3. We follow the same evaluation methodology as the one described in Section 4.4.2. Note that all the algorithms have been implemented using the *Apache Lucene* search engine. All the non-personalized approaches used as baselines are described in Table 5.4.

In the following, Section 6.5.1 provides answers to question 1, Section 6.5.2 addresses question 2 and 3, Section 6.4 gives a discussion on question 4, Section 6.6.1 shows the analysis of question 5, and finally, Section 6.6.2 tackles question 6 based on different criteria.

6.5.1 Performance Comparison

In this section, we compare all the personalized ranking functions with respect to the results' quality, while highlighting SoPRa. The results of the comparison are illustrated in Figure 6.2, while varying γ.



(a) Mean Average Precision

(b) Mean Reciprocal Rank

Figure 6.2 – Comparison with the baseline while varying γ.

6.5.1.1 SoPRa vs non-personalized ranking approaches

The results show that SoPRa and its extension are much more efficient than all the non-personalized approaches for all values of γ. Hence, we conclude that the personalization efforts introduced by SoPRa improve the search quality. We also notice that most of the non-personalized approaches decrease their performance for high values of γ. This is certainly due to the fact that they are not designed for personalized

search, since these approaches fail in discriminating between users in spite of their preferences.

6.5.1.2 SoPRa vs personalized ranking approaches

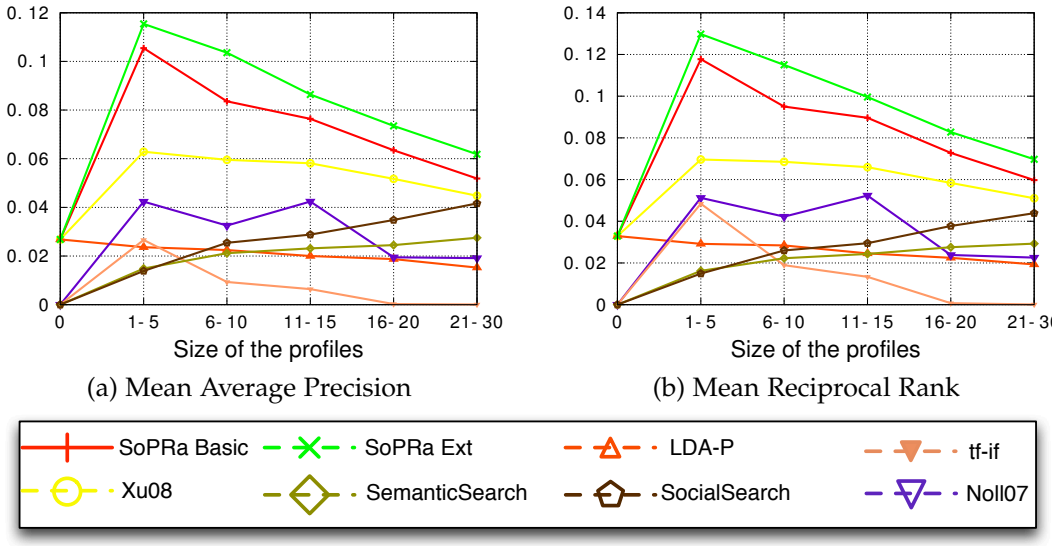
Here, the obtained results also show that SoPRa is much more efficient than all the personalized ranking functions for all values of γ. Especially, SoPRa outperforms the LDA-P approach and the Xu08 approach. We also notice that the Noll07 and the tf-if approaches give poor results. This is certainly due to the fact that they fail in ranking documents that doesn't share tags with users, since in our experiment we remove the triplets that associate the user, the query terms and documents. We also notice that the performances of LDA-P decrease for high value of γ. This is certainly due to the fact that the basic LDA approach fails in modeling the topic of documents based on their annotations. Adapting this approach for leveraging social information could be part of our future work.

Finally, we note that the better performance are obtained for $\gamma \in [0.6, 0.8]$ for the basic SoPRa, a compromise between the user interest matching score and the query affinity matching score. As for the extended SoPRa, it seems that γ has no impact on the results. This show that the extension proposed takes full advantage of the user interest matching score and the query affinity matching score. We also note that the extension of SoPRa provides better performance than the basic one. This shows that considering users individually and their similarities to the query issuer provide a better estimation of the relevance of documents. We also note that many non-personalized approaches are more efficient than personalized approaches, e.g. BL-Q and SPR, which outperform Noll07, tf-if, and LDA-P.

6.5.2 Performance on Different Users

In this section, we study the ability of the personalized ranking functions to achieve good performance for users that have different profile length, i.e. users who used few terms in their tagging actions. Hence, we propose to compare these approaches using the evaluation process described in Section 4.4.2. We select 2000 query pairs (u, t) based on the number of tags the users used in their tagging actions. The query pairs are grouped into 6 classes: "0", "1-5", "6-10", "11-15", "16-20", and "21-30", denoting how many tags users have used in their tagging actions, e.g. class "1-5" is composed by users who have a profile length between 1 and 5. Note that we fixed γ to 0.5 for all the approaches. The experimental results are shown in Figure 6.3.

The results show that the performance of all the profile based approaches decreases for users with high profile length, i.e. SoPRa, Xu08, LDA-P, Noll07, tf-if. This is certainly due to the fact that these approaches fail to determine the user expectations, if he expressed his interest in different fields. However, the affinity based personalization approaches increase their performance for users with high profile length. These approaches are based on other user experiences with common tastes



(a) Mean Average Precision (b) Mean Reciprocal Rank

Figure 6.3 – Performance comparison on different queries, while fixing $\gamma = 0.5$.

and affinities with the query issuer for ranking documents. Hence, we believe that modeling a user profile with simply his tags is not enough to generate satisfactory search results, especially for active users on social networks. We must go beyond that by considering their social relatives for ranking purpose. This is part of our future work for improving SoPRa.

Finally, we note that many personalized ranking functions are not able to provide a suitable ranking of documents for users with no tags. Currently, all the approaches that are able to rank documents for users with no tags relay on the Lucene naive score for dealing with cold start problem.

6.6 User Survey

As done for the PSDR framework in Chapter 5, we also evaluate SoPRa using an end-user survey. We follow the same evaluation methodology, where we selected 335 pairs of queries and users. These users are considered as query initiators and have used all the selected query tags at least once on the same document. We then ran the queries using our approach and the baselines that performed the best in the offline evaluation, i.e. at each iteration, the user is presented with lists of ranked documents using (i) our approach and (ii) a randomly selected baseline algorithm. Each list contains seven documents. Note that for all approaches, γ was set to 0.5.

In the assessment phase, 19 volunteers participated to judge the relevance of the results. Each volunteer (who is considered as a query initiator) was shown, in addition to the results for the query from the pool: (i) the documents from the query initiator that contain at least one of the query tags and (ii) the tags he used in his tagging

actions. This is to help the volunteers to understand the personal context of the (real) query initiators as well as their interests. This way, we intend to overcome the aforementioned problem of subjectively assessing result qualities with the eyes of the query initiator.

Once a list is presented to a participant, he/she marks each result as: very relevant, relevant, or irrelevant w.r.t the context of the (real) query initiator. This process is performed by the evaluator without knowing which algorithm generated the result. Figure 5.12 shows the interface that the users obtained when they participated to the survey. This interface contains (i) the tags the user used in his tagging action (in the top right part), (ii) the documents he tags with the query terms (in the right part), and (iii) the two lists of results to be judged after the query was issued.

The quality of each result was measured by the normalized discount cumulative gain (nDCG@7) and by precision at 10 (P@7), averaged over the set of judged queries. For DCG calculation, we used gains (2,1,0) for the three relevance levels respectively, and the discount function used was is given in Equation 2.6. Normalization (nDCG) was done by dividing the DCG value with an ideal DCG value calculated as all results are highly relevant. For P@7 calculation, we considered any positive judgment as relevant. The obtained results are shown in Figure 6.4 as measured by NDCG@7 and P@7.

The main outcome of this survey can be summarized as follows: (i) this user survey confirms, to some extent, the results obtained in the offline evaluation regarding the Basic SoPRa. (ii) The extended SoPRa generated results of poor quality regarding the other approaches. This is probably due to the fact that it hasn't been evaluated on a large number of queries, e.g. Extended SoPRa versus Xu08 have been evaluated on only 2 queries, which is not really significant. (iii) The advantage observed by SoPRa is not as important. As in the previous survey, several participants mentioned the difficulty in judging the relevance of the queries, mostly because of unfamiliarity with the users they are related to. Finally, (iv) the evaluation of the extended version of SoPRa should be reinforced by collecting more users' judgement. This is to have a clear statement regarding its performance.

6.6.1 Efficiency Analysis

We compare here the algorithms from the point of view of complexity.

In Xu08 approach, four parameters are used by two functions: (i) a cosine similarity $Cos(\overrightarrow{p_u}, \overrightarrow{T_d})$ computes an inner dot product between user weight vector $\overrightarrow{p_u}$ and document weight vector $\overrightarrow{T_d}$, with a complexity of $O(\mid \overrightarrow{p_u} \mid)$, if we assume that $\overrightarrow{p_u}$ is the smallest vector of the two, and (ii) $Sim(\overrightarrow{q}, \overrightarrow{d})$ computes the textual matching score between d and q, with a complexity of $O(\mid \overrightarrow{q} \mid)$ in the case of a similarity based on cosine similarity. The global complexity of this approach is then $O(\mid \overrightarrow{p_u} \mid + \mid \overrightarrow{q} \mid)$ for each result of a query.

LDA-P approach relies on an LDA-generated model in order to compute a ranking.

This implies a heavy preprocessing algorithm that is executed once, at the start of the system lifetime, and whose execution can be resumed from time to time to update the system model. That said, the cost of each query is estimated as follows: the skeleton of the algorithm is the same as for Xu08, except the two first parameters (vectors $\overrightarrow{p_u}$ and $\overrightarrow{T_d}$ of Xu08) are replaced with two other vectors, namely $\overrightarrow{u_{topic}}$ and $\overrightarrow{d_{topic}}$, (topic weight vectors for respectively a user and a document) that are also pre-indexed by LDA and have constant time access for their associated weights. Hence, the global complexity is $O(n + \mid \overrightarrow{q} \mid)$ where $n = \mid \overrightarrow{u_{topic}} \mid = \mid \overrightarrow{d_{topic}} \mid$.

The basic SoPRa is based on Xu08, but adds a supplementary term $Cos(\overrightarrow{q}, \overrightarrow{T_d})$ to take into account the social proximity between the query and the document. The cost of this added term is $O(\mid \overrightarrow{q} \mid)$ if we assume that the query is shorter than the document tag list. Hence, the global complexity of SoPRa becomes $O(\mid \overrightarrow{p_u} \mid + 2 \times \mid \overrightarrow{q} \mid)$.

As for the extended version of SoPRa, since it takes into account each user who tags a document individually, its complexity is pretty different. The global complexity is given by $O(\mid U_d \mid \times 2 \times \mid \overrightarrow{p_u} \mid + \mid U_t \mid \times (\mid \overrightarrow{p_u} \mid + \mid \overrightarrow{q} \mid) + \mid \overrightarrow{q} \mid)$. We consider this complexity as quadratic.

Noll07 and tf-if have been summarized into a sum over the tags of the set $T_u \cap T_d$. The size of this subset does not exceed that of the smallest between T_u and T_d. Hence, the global complexity of these two approaches for computing the score of a document can be summarized to $O(\mid T_u \mid)$ if we assume that $\mid T_u \mid \leq \mid T_d \mid$.

The first affinity-based approach, SemanticSearch, computes two successive sums over the terms of the query and the users who used these terms. The complexity of this ranking function for computing the score of a document is then estimated to $O(\mid q \mid \times \mid U_t \mid)$.

The second affinity-based approach, SocialSearch, looks like SemanticSearch, but with an added cosine similarity computation for each term of the sum and a different user who used this term. This cosine similarity has a cost of $O(\mid \overrightarrow{p_{u_i}} \mid)$. The global complexity is then $O(\mid q \mid \times \mid U_t \mid \times \mid \overrightarrow{p_u} \mid)$.

In summary, if we look at the complexity of each algorithm, we can distinguish 3 categories of algorithms, upon which we find common properties in term of computing complexity. First, as a common property, each algorithm relies more or less on the size of the user profile $\mid \overrightarrow{p_u} \mid$, which is the same as the cardinal of the tag set of a user, $\mid T_u \mid$, except SemanticSearch approach that does not use this value. Then, 3 categories of complexity emerge: (i) Xu08-based algorithms are the three first approaches, (ii) follow the Noll07 and tf-if algorithms, and (iii) the two last Affinity-based algorithms conclude the test. The second category of algorithms, is the most efficient with a complexity borned by the profile size of the user. Xu08-based algorithms come second in complexity, keeping the user profile size linearity and adding to it the query length. Finally, the third category, Affinity based algorithms, is the slowest one, because these ones grow with at least the product of the profile size and the query size.

6.6.2 Summary

Table 6.2 summarizes the personalized ranking functions studied from different point of views. This table is built upon our appreciation of the approaches.

Table 6.2 – Summary of the analysis.

	General Performance[a]	Time Complexity[b]	Cold Start[c]	Adaptability[d]	Effectiveness[e]
Xu08	**	$O(\mid \overrightarrow{p_u} \mid + \mid \overrightarrow{q} \mid)$	+	+	-
LDA-P	*	$O(n+ \mid \overrightarrow{q} \mid)$	+	-	-
SoPRa Basic	***	$O(\mid \overrightarrow{p_u} \mid +2\times \mid \overrightarrow{q} \mid)$	+	+	-
SoPRa Ext	***	$O(\mid U_d \mid \times 2\times \mid \overrightarrow{p_u} \mid + \mid U_t \mid \times (\mid \overrightarrow{p_u} \mid + \mid \overrightarrow{q} \mid)+ \mid \overrightarrow{q} \mid)$	+	+	-
Noll07	*	$O(\mid T_u \mid)$	-	+	-
tf-if	*	$O(\mid T_u \mid)$	-	+	-
SemanticSearch	**	$O(\mid q \mid \times \mid U_t \mid)$	+	+	+
SocialSearch	**	$O(\mid q \mid \times \mid U_t \mid \times \mid \overrightarrow{p_u} \mid)$	-	+	+

[a] The general retrieval performance of the approaches. *** : very effective; ** : effective; * : not effective.

[c] The cold start is a potential problem of systems to handle effectively new entities, e.g. users, items, or tags. In other words, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information. + : can cope with cold start problem; - : cannot cope with cold start problem.

[b] The complexity is given for computing the ranking score of one document.

[d] Adaptability refers to the ability of approaches to consider new data and to quickly update their model. Considering new data is a key problem for these ranking functions since they are based on social networks which are growing quickly with the intense activity of users by commenting, publishing, sharing content and explicitly express their opinion. + : can easily update the model; - : cannot easily adapt the model.

[e] The effectiveness of the approaches for different profile lengths. + : effective for users with high profiles lengths; - : not effective for users with high profiles lengths.

As a conclusion, we believe that SoPRa is the ranking function that offers the best trade off between retrieval performance, time complexity, cold start problem, and adaptability. However, the retrieval performance of this approach decreases for users with high profile length. We believe that we can tackle this issue by extending this ranking function by leveraging the social relatives of the query issuer. This is

part of our future work. Moreover, we believe that we can improve the cold start problem of this ranking function by considering social relevance of documents. Social relevance refers to information socially created that characterizes a document from the point of view of its interest, i.e. its general interest, its popularity, etc. SPR could be a good algorithm, since it is currently outperforming the Lucene naive ranking function (which is currently used for addressing the cold start problem in SoPRa).

6.7 Conclusion and Future Work

This chapter discussed a contribution to the area of Social Information Retrieval (SIR), especially Social Web Search. In this context, many approaches have been proposed to improve the ranking process by personalizing it using social features. We reviewed many of these personalized ranking functions and we proposed a new one called SoPRa. We tried to mainly answer the following questions: *What are these functions and how do they work? What is the context where each function is more efficient? What is the best ranking function? To which extent are they efficient compared to non-personalized approaches?*

To address these questions, we proposed: (i) a study of the state of the art ranking functions in social collaborative setting, (ii) an analysis of the performances of these personalized social ranking functions and a comparison with non-personalized social approaches, and (iii) a discussion on the effectiveness, the weakness and the performance of each approach in different contexts.

As a conclusion, SoPRa presented the best characteristics in terms of retrieval performance, time complexity and adaptability. However, many improvements remain possible for improving this approach. Those include: (i) considering the social relevance score of documents, (ii) considering the social relatives of users, and (iii) considering the temporal dimension of social users' behavior e.g. considering the evolution of the taste of users in the ranking function.

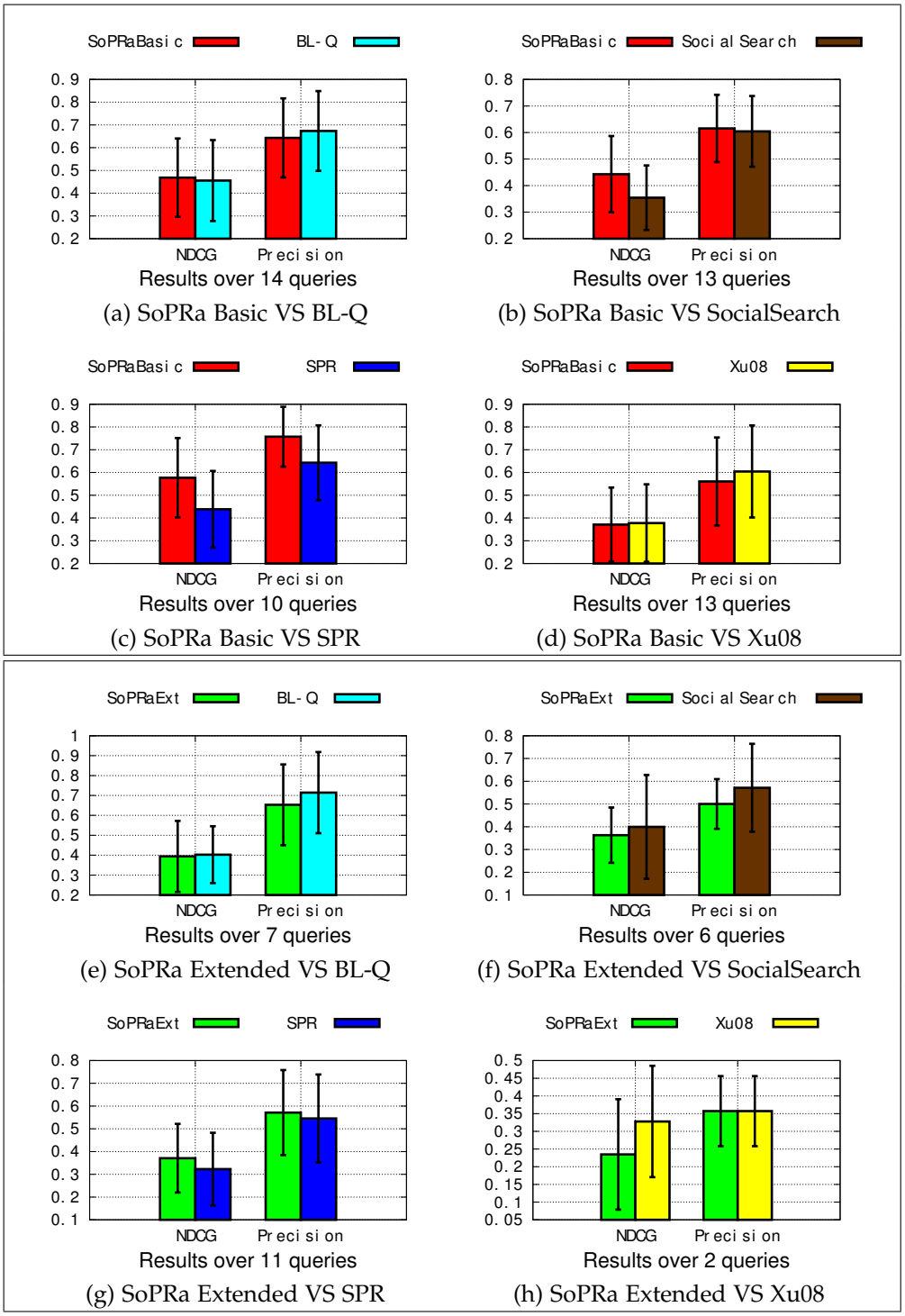


Figure 6.4 – Results of the SoPRa user survey: The precision of the search results for different algorithms measured by nDCG@7 and P@7.

Chapter 7

LAICOS: Towards A Personalized Social Web Search Engine

7.1 Introduction

We presented in the previous three chapters three approaches that aim to leverage the social dimension of the Web, for improving the IR process. In addition to put into practice these features and algorithms, we implemented a social Web search engine called LAICOS, which put into practice these algorithms. LAICOS is an open source prototype, in which social information and personalization are at the heart of the IR process. In its current implementation, this prototype is relying on social bookmarking systems as a source of social information for personalizing and enhancing the IR process. However, it can be extended to use any source of social metadata, i.e. tweets, comments, etc.

The current chapter describes details of the design and implementation of LAICOS. In Section 7.2, we describe the logical architecture of LAICOS. Then, in Section 7.3, we present the graphical user interface (GUI) of LAICOS, and how it can be used to fully leverage its functionalities. In Section 7.4, we present how LAICOS can be adapted to use other sources of social information, and a generalization of our algorithms. Finally, we conclude and give some future work in Section 7.5.

7.2 Architecture of LAICOS

Figure 7.1 illustrates the main components of LAICOS, which are: (i) a set of connectors, crawlers, and a database storing the data, (ii) a data indexing engine, and (iii) a query processing engine.

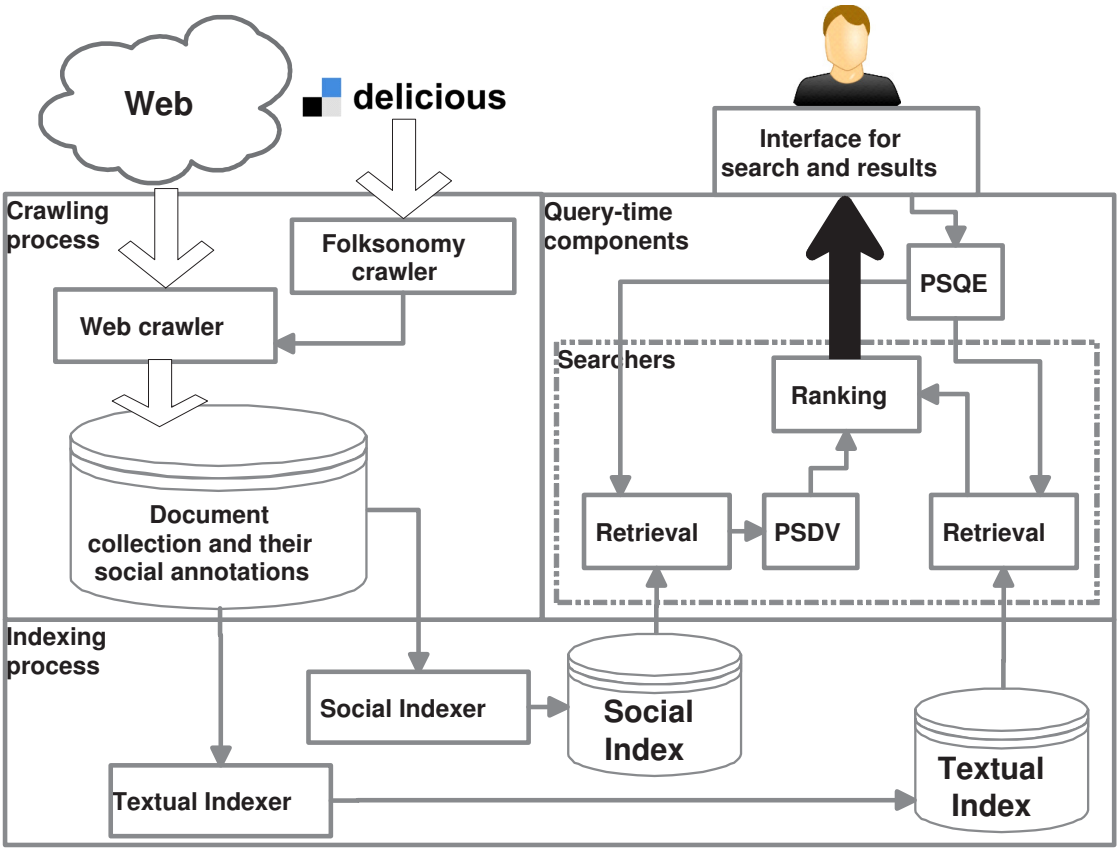


Figure 7.1 – Architecture of LAICOS

7.2.1 Crawlers in LAICOS

A crawler refers commonly to a tool, which browses the Web for downloading entities/objects, e.g. Web pages, images, etc. Hence, a crawler in a search engine is an essential piece that should balance coverage (i.e. volume) and quality (i.e. freshness) [BYRN11]. We designed LAICOS in such a way, it possesses the following two types of crawlers:

(i) A crawler for Web pages based on *Heritrix* [34], which was specifically designed for Web archiving and crawling.

(ii) A folksonomy crawler engine, which will download all the annotations that are associated to Web pages through APIs [35]. Annotations are recovered from social bookmarking systems, especially the *delicious* website. Other sources can be connected to this crawler.

All the downloaded documents and annotations are stored into a database for being indexed.

34. http://crawler.archive.org/index.html
35. http://delicious.com/developers

7.2.2 Social Indexes in LAICOS

Crawled Web pages and their social annotations are stored into a repository. Two indexing engines are responsible for indexing and keeping up to date the following index structures:

(i) *A textual content-based index* structure, which is based on indexing the collection of crawled documents using the *inverted index* structure. The *Apache Lucene* search engine is handling this task [36]. This *textual content based index* structure is well described in [MHG10].

(ii) A *social-based index* structure, which is based on the crawled annotations assigned by users to Web pages in social bookmarking websites. We implemented our own indexing engine and structure for this task.

In this section, we only describe the *social-based index* structure of LAICOS since it is our own proposal and contribution. This index consists of the following seven main data structures (See Table 7.1 for the internal format and compression used for each structure):

— **Docs**: it stores Web pages' ids (md5 hash of a Web page name), the number of tags and users associated to the Web page, as well as the offset in the Docs_Users posting list.

— **Tags**: it stores tags' ids (md5 hash of the tag text), the number of Web pages and users associated to the tag, and the offset in the Tags_Docs posting list.

— **Users**: it stores users' ids (md5 hash of the user user name), the amount of Web pages and tags associated to the user, and the offset in the Users_Tags posting list.

— **Docs_Users**: it stores the posting list of users for Web pages. In particular, for each Web page, this structure stores: the id of the user who tags this Web page, the amount of tags he has used to annotate this Web page, and the offset in the Bookmarks posting list.

— **Tags_Docs**: it stores the posting list of Web pages for tags. In particular, for each tag, this structure stores: the id of the Web page which is tagged with this tag and the amount of users who have used this tag to annotate this Web page.

— **Users_Tags**: it stores the posting list of tags for users. In particular, for each user, this structure stores: the id of the tag used by this user and the amount of Web pages tagged by this user with this considered tag.

— **Bookmarks**: it stores the posting list of tags for a document and a user. In particular, for each unique pair of Web page and a user, this structure stores: the ids of tags used by this user to annotate this Web page.

36. http://lucene.apache.org/core/

Structure	***Contents***	***Size***
Docs 44 Bytes	*id*	32 Bytes
	Number of users who annotate this Web page	4 Bytes
	Number of tags used to annotate this Web page	4 Bytes
	Byte offset in Docs_Users	4 Bytes
Tags 44 Bytes	*id*	32 Bytes
	Number of users	4 Bytes
	Number of Web pages	4 Bytes
	Byte offset in Tags_Docs	4 Bytes
Users 44 Bytes	*id*	32 Bytes
	Number of Web pages	4 Bytes
	Number of tags	4 Bytes
	Byte offset in Users_Tags	4 Bytes
Docs_Users 40 Bytes	*userid*	32 Bytes
	Number of tags	4 Bytes
	Byte offset in bookmarks	4 Bytes
Tags_Docs 36 Bytes	*docid*	32 Bytes
	Number of users	4 Bytes
Users_Tags (36) 36 Bytes	*tagid*	32 Bytes
	Number of Web pages	4 Bytes
Bookmarks 32 Bytes	*tagid*	32 Bytes

Table 7.1 – Details on the format and compression used for each index data structure.

For more clarity, Figure 7.2 presents a graphical illustration of the architecture of the social index that we designed for LAICOS. It also shows the existing links and pointers between offsets of its files. Note that the size of the social index structure of the *delicious* dataset described in Table 4.3 is about 811 Megabytes.

7.2.3 Query Pre-processing Engine in LAICOS

In IR systems, queries are usually pre-processed by being reformulated. This process includes either: (i) the reduction of queries, which is a technique to reduce long queries to more effective ones [KC09], or (ii) expansion of queries, which consists of enriching the user's initial query with additional information [Eft96].

In LAICOS, queries are interpreted and processed using the Personalized Social Query Expansion (PSQE) framework, discussed in Chapter 4. We recall that in order to achieve social and personalized expansions of a query, PSQE considers: (i) the semantic similarity between candidate terms and the query, and (ii) the extent to which the candidate terms are likely to be interesting to the user.

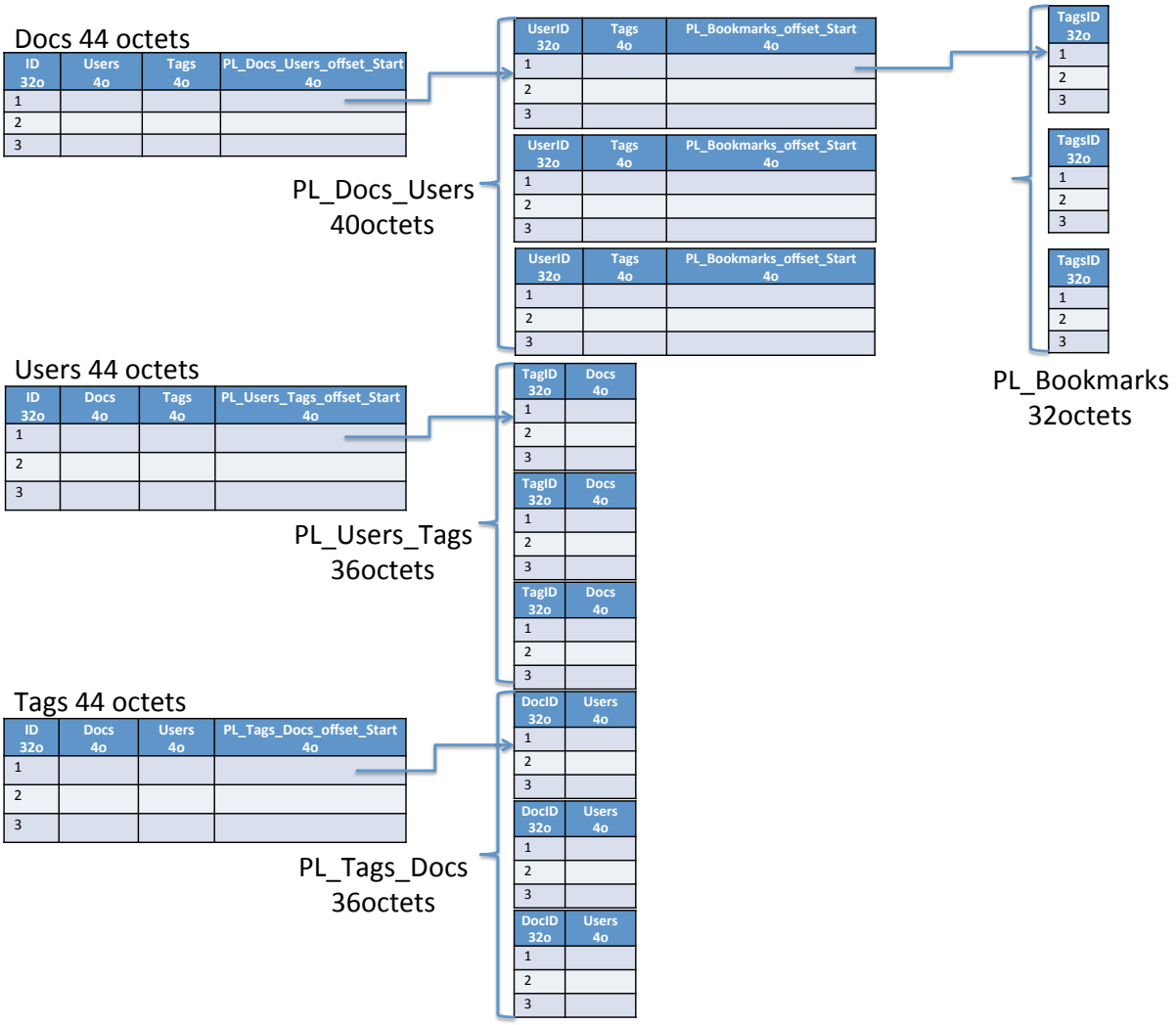


Figure 7.2 – Graphical representation of the architecture of the social index of LAICOS.

7.2.4 IR Models in LAICOS

Here, we mean by an IR model the definition of a conceptual model to represent documents and queries [BYRN11]. In LAICOS, entities are modeled as follows:

(i) To model the textual content of documents, LAICOS is based on the *Apache Lucene* search engine. This latter follows the Vector Space Model and the Boolean model [MHG10].

(ii) To model users' profiles, LAICOS follows the Vector Space Model introduced in Section 2.2.2 and the definition given in Section 2.2.6.

(iii) To model the social annotations associated to documents, LAICOS uses the Personalized Social Document Representation (PSDR) framework introduced in Chapter 5.

7.2.5 Ranking Model in LAICOS

In LAICOS, the ranking score for a document d that appears in the results list obtained when a user u issues a query q is computed using the Social Personalized Ranking function SoPRa introduced in Chapter 6. We remind that the ranking score is computed as follows:

$$Rank(d,q,u) = \gamma \times Sim(\overrightarrow{q}, \overrightarrow{S_{d,u}}) + (1-\gamma) \times \left[\beta \times Sim(\overrightarrow{p_u}, \overrightarrow{S_{d,u}}) + (1-\beta) \times Sim(\overrightarrow{q}, \overrightarrow{d})\right] \tag{7.1}$$

where, γ and β are weights that satisfy $0 \leq (\gamma, \beta) \leq 1$, Sim is the cosine similarity measure between the different vectors computed using Equation 2.3, $\overrightarrow{S_{d,u}}$ is the PSDR of the document d according to the user u, and $\overrightarrow{p_u}$ is the profile of u. In the next section, we briefly describe how a user query is processed in LAICOS.

7.3 Lifecycle of a User Query

The on-line IR sub-process which is illustrated in the right part of Figure 7.1 takes in charge the user query in LAICOS. In the following, we describe a scenario that will illustrate how to use LAICOS in order to fully leverage its functionalities.

1. In order to fully exploit the potential of LAICOS, the user is able to register his *delicious* account, which is instantly crawled. This will serve to construct the user profile, in order to provide a personalized search service.
2. The user can choose different configurations of the parameters of the LAICOS social search engine in order to fix them based on what are his expectations, e.g. does he want personalization? what is the degree of personalization he want? how many closest users he want to use to make personalization? etc. This GUI is depicted in Figure 7.4, where for each parameter, an explanation is given.
3. The user issues a query in the form of keywords using the main interface illustrated in Figure 7.3, which is similar to the one of a classical search engine.
4. The query is handled by the query processing engine of PSQE, which, by default, includes the removal of English stop-words and the application of the Porter's stemming algorithm [Por97]. The user is also able to enable and tune the parameters of the expansion process of PSQE through the system interface as illustrated in Figure 7.4.
5. As an output of this first step, PSQE returns an adapted user query as a vector of weighted terms, which is expected to be as close as possible to the user' information needs.
6. The new query is processed by a retrieval engine, which retrieves all documents that contain the query terms in their textual content. This process is based on the *Apache Lucene* search engine.

Figure 7.3 – LAICOS Homepage

7. For each document, a Personalized Social Document Representation is built using the PSDR framework. Again, the PSDR framework can be tuned through the system interface as illustrated in Figure 7.4.
8. A ranking score is computed for each retrieved document using the SoPRa function as described in Section 7.2.5.
9. The resulted list of documents is sorted based on this final ranking score from the most relevant to the less relevant one.
10. Finally, the top ranked documents are formatted for presentation to the user as illustrated in Figure 7.5. A summary of the terms that form the PSDR of each document is given. The user can use these terms in order to tag these Web pages.

7.4 Generalization and Extension

In this section, we discus the possibility of LAICOS to consider other sources of social information. Indeed, there are diverse sources of heterogeneous social information, each of which contains valuable knowledge that can be reused to enhance the IR process. Hence, we propose to generalize the LAICOS social Web search engine in general, and our algorithms in particular, in order to:

(i) exploit various types of social information that its metadata is relevant to the semantic content of Web documents and users. Such social metadata extensively include anchor text, search query, social annotation and so on.

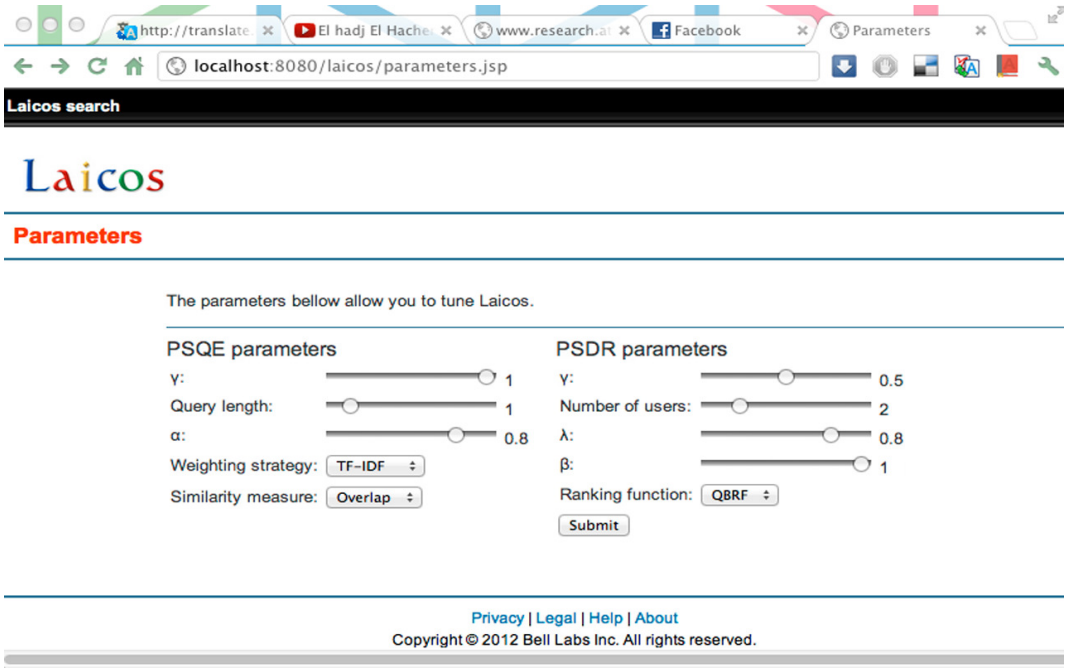


Figure 7.4 – Parameter settings

(ii) adapt to extensive types of Web documents provided that the documents have initial textual terms and/or initial social annotations. Such applicable documents range from Web pages, Web images, Web videos to traditional text documents.

(iii) generalize to more types of social data for enhancing their content concurrently.

Throughout this thesis, we focused mainly on the use of folksonomies to enhance the IR process. Folksonomies represent tagging actions that are modeled using tripartite relations, which link users, tags and resources. However, there exist other social activities, which are modeled using other relations, e.g. bipartite relations. In the following, we show how LAICOS can leverage other social relations and activities through some transformations.

7.4.1 Transformation to a Tripartite Graph

Many social networks allow users to share and comment entities on the Web. For example, on most online news sites (e.g. New York Times), each article is accompanied by buttons corresponding to Facebook, Twitter, etc., which allow a user to quickly post and comment the article to his favorite social site and share it with friends. This forms a ternary relation between the user, the article, and the comment. Hence, as illustrated in Figure 7.6a, through some textual processing (e.g. remove stop words and stemming) and graph transformations, we can end up with a pseudo-folksonomy, which link the user, the document, and terms. Then, this pseudo-folksonomy is used as input for LAICOS and our algorithms.

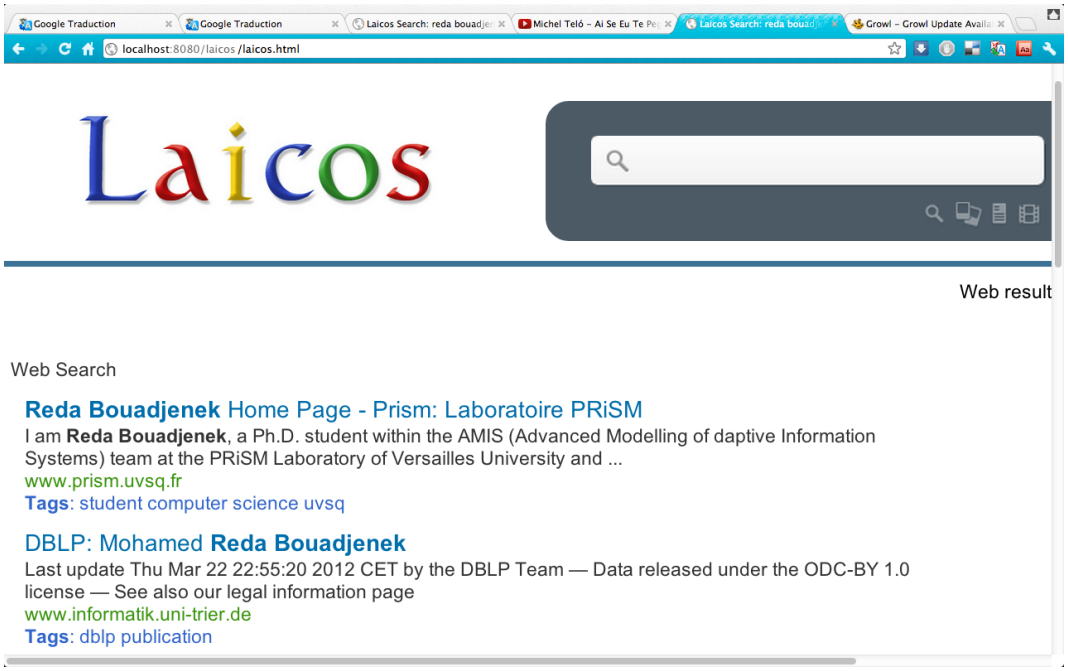


Figure 7.5 – Search results

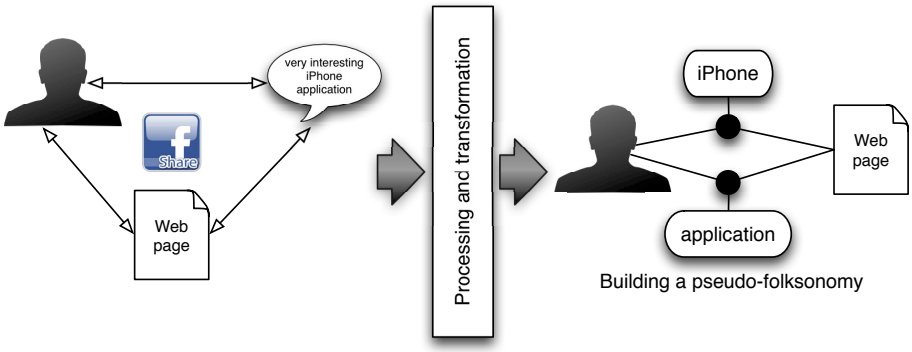
7.4.2 Transformation to a User-Term Bipartite Graph

In many social networks, users are able to express their opinions and interests through posted messages, without being explicitly associated to articles or documents, e.g. post a tweet, or status on Facebook. Hence, as illustrated in Figure 7.6b, through some textual processing (e.g. remove stop words and stemming) and graph transformations, we can end up with a User-Term bipartite graph, which links users and terms. This graph can be easily reused to enhance users' profiles, used in LAICOS for personalizing the results.

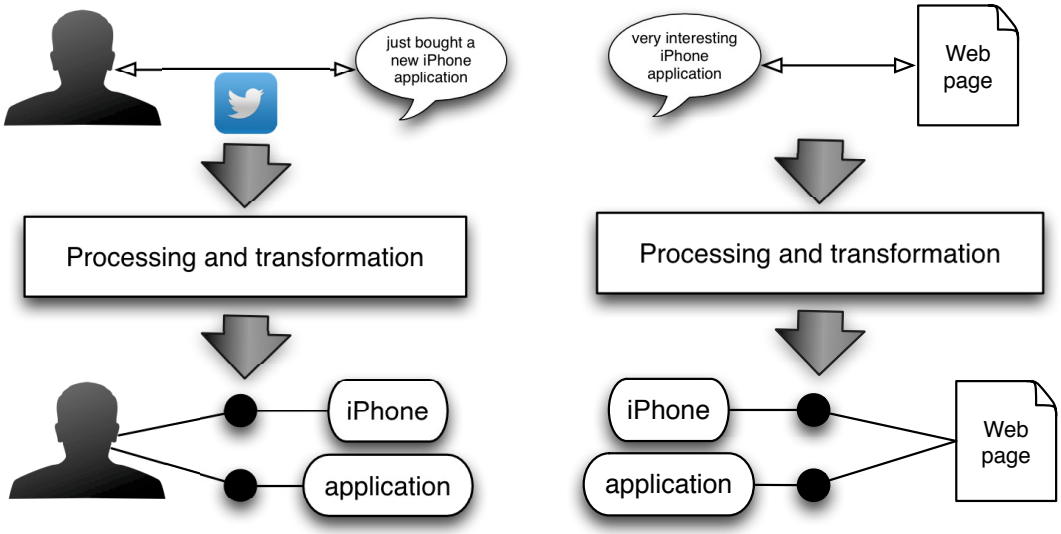
7.4.3 Transformation to a Term-Doc Bipartite Graph

In many social activities, users can anonymously comment or annotated Web pages and articles. This information can be seen as metadata that describe the content of these entities. Hence, as illustrated in Figure 7.6c, through some textual processing (e.g. remove stop words and stemming) and graph transformations, we can end up with a Term-Doc bipartite graph, which links terms and documents. This graph can be easily reused to enhance the content of documents and its representation for using SoPRa, and enrich the graph of Tags constructed in the PSQE framework.

Finally, in this section we presented and discussed some ways to generalize LAICOS and our algorithms to consider other types of social data. We believe that it is necessary to handle these sources of social data in LAICOS in order to fully leverage the social dimension of the Web. However, currently, we didn't formalize and implement algorithms for that.



(a) Transformation to a tripartite graph.



(b) Transformation to a User-Term bipartite graph.

(c) Transformation to a Term-Doc bipartite graph.

Figure 7.6 – Generalization models to LAICOS.

7.5 Conclusion

In this chapter, we introduced LAICOS, a personalized social Web search engine that considers the social context of both users and documents in the IR process. On the one hand, the social context of documents is added as a layer to the textual content traditionally used to index a collection of documents to provide a personalized social representation of documents using the PSDR framework. On the other hand, the social context of users is used for profiling users, and providing personalized search results through the Personalized Social Query Expansion framework (PSQE) and the Social Personalized Ranking function (SoPRa). The prototype of LAICOS is implemented using *Apache Lucene* IR platform and preliminary experiments are conducted on documents extracted from *delicious*.

Finally, LAICOS is a platform that show the transition from theory developed in

this thesis to application and engineering. It also show the commissioning of all the algorithms and approaches developed throughout this thesis. However, many work remains to be done in LAICOS. For instance, this includes mainly the generalization part discussed in Section 7.4. This include implementing and developing algorithms for leveraging diverse sources of social information.

Chapter 8

CONCLUSION

Information Retrieval and Social Networks Analysis are still a growing research area in computer science. Specifically, classic models of IR and even the IR paradigm are about to evolve with the socialization of the Web. In this context, this thesis investigated the problem of leveraging the social dimension of the Web, in order to adapt, enhance, and improve the classic IR process. Below, Section 8.1 presents a summary of our contributions and then Section 8.2 presents the possible future directions of our research work.

8.1 Contributions

Nowadays, we estimate the Google's index size to approximatively 50 billion of Web pages [37]. This gives us an illustration of how big is the Web (without considering the Deep Web [MKK+08]). This huge amount of Web pages makes it impossible for users to browse all these documents in order to find relevant information. Information retrieval comes then as a mean to assist users in finding relevant information on the Web. Since the Web was initially considered as static and expected to evolve by the contribution of few users, i.e. authors of Web pages, a large number of IR models and techniques have been developed during its first 15 years, which go in this sense. However, with the emergence of social platforms such as Facebook and Twitter, all users are able to contribute to enrich the Web and produce content. This evolution of the Web is known as Web 2.0 or Social Web. In this context, this PhD thesis investigated the way IR can fully leverage the socialization and interaction between entities in the social Web. The ultimate goal still always to retrieve all documents that are relevant to a user's query, while retrieving as few irrelevant documents as possible.

Hence, we first started by studying the state of the art in Social Information Retrieval, the topic that bridges the gap between IR and social networks. We proposed a taxonomy for SIR models based on the fundamental affinities that exist between the tools and approaches studied. We categorized SIR models into three main classes:

37. http://www.worldwidewebsize.com/

(i) *Social Web Search*, in which social information is used in order to improve the classic IR process, e.g. documents re-ranking, query rewriting, user profiling, etc.

(ii) *Social Recommendation*, in which the user's Social Network is used to make recommendation, e.g. using a social trust network [MYLK08].

(iii) *Social Search*, in which it is a matter of finding information with the assistance of social resources, such as by asking friends, reference librarians, or unknown persons on-line for assistance [MTP10].

In this PhD thesis, we focused on *social Web search* for adapting and improving the classic IR process. We proposed algorithms and approaches to improve the following three tracks: (i) query reformulation using extra knowledge, i.e. expansion of the user query, (ii) improvement of the IR model, i.e. the way documents and queries are represented and matched to quantify their similarities, and (iii) post filtering or re-ranking of the retrieved documents (based on the user profile or context).

In our first contribution, we tackled the problem of query expansion using social knowledge. Basically, this social knowledge is the users' vocabulary extracted from social tagging platforms. The proposed approach for expanding users' queries consists in three main steps:

(i) Determining similar and related tags to a given query term through their co-occurrence over resources and users. This step extracts semantics from the whole social graph of a Folksonomy without losing information, i.e. by exploiting the co-occurrence of tags over resources and users. This step led to the creation of a graph of tags, where edges represent semantic relations between tags. This graph is further used to extract terms that are semantically related to a given term of a query to perform the query expansion.

(ii) Constructing a profile of the query issuer based on his tagging activities, which is maintained and used to compute expansions. These profiles are used to estimate users' interests in order to personalize the expansion of queries.

(iii) Finally, expanding the query terms, where each term is enriched with the most interesting tags based on their similarities (semantic similarity between terms) and their interest to the user (similarity to the user's profile).

The results obtained as for the offline evaluation of our approach show significant improvement of the results' quality compared to the closest state of the art methods. However, these results should be reinforced by an online evaluation to get feedback from end users.

Next, we investigated the problem of IR modeling using social information that comes from tagging systems. To this end, we proposed the Personalized Social Document Representation framework (PSDR). Basically, for a given document that match a particular query, the PSDR framework acts as follows:

(i) Representing the document using a Users-Tags matrix. This matrix is first sized by selecting relevant users (to both the query issuer and the document), then

it is weighted for estimating the extent to which, each user thinks that a tag is associated to the considered document.

(ii) Each row i in the Users-Tags matrix of a given document represents the personal representation of the user u_i. This matrix is expected to be sparse, since it contains many missing values that should be inferred to enhance the PSDR. Hence, a matrix factorization process is used to infer the PSDR of the considered document to the query issuer based on identifying weighting patterns.

(iii) Finally, we compute a ranking score for the document based on its PDSR and its textual content. We used a linear function in order to merge the two parts and produce a final ranking score.

The PSDR framework has been evaluated both offline and online, i.e. user survey. The obtained results in the offline evaluation showed a considerable improvement with respect to the closest state of the art methods. As for the online evaluation, the obtained results was quite disputable, mostly because participants were troubled for assessing the results' quality. A computational analysis of the PSDR framework shows that its complexity scales linearly with the number of retrieved documents that match the query.

Then, for ranking search results, we first proposed a state of the art on personalized social ranking functions. Then, based on the technical weakness of these ranking functions, we proposed a **So**cial **P**ersonalized **Ra**nking function called SoPRa. SoPRa leverages the social dimension that surrounds both documents and users. In its basic form, SoPRa ranks documents according to: (i) a textual content matching score of documents and the query, (ii) a social matching score of documents and the query, and (iii) the social interest score of the user to documents. Also, we proposed an extended version of SoPRa, which considers each user, who annotates a Web page individually, in order to fully leverage the collaborative setting, i.e. similarity and proximity between users, trustworthiness and confidence of users, etc.

We also performed an intensive evaluations on SoPRa to estimate its effectiveness compared to the closest state of the art methods. These assessments also include a complexity analysis, an offline and an online evaluations. The results obtained in the offline assessment phase showed significant improvement of the results's quality. However, since we didn't collect enough feedback from users in the user survey, the obtained results was not so important. The obtained results also showed possible improvements, that we can bring to SoPRa, i.e. profiling users with not only the terms they commonly used in their tagging actions.

Finally, as a last contribution, we proposed LAICOS, a social Web search engine that is based on all the algorithms developed and discussed throughout this thesis (including the state of the art methods). LAICOS is an open source platform that can be used as a search engine to make daily information search activities. It can also be used to develop SIR algorithms. Furthermore, it allows to (i) easily compare their retrieval performance to the closest state of the art methods, and (ii) easily extend the existing library of algorithms and methods for the research community. This is to

help researchers in the task of assessing the quality of their contributions and comparing them to other approaches, preferably under the same conditions. Currently, the prototype contains a library of different SIR algorithms and techniques related to the different levels of the IR process. The prototype relies on social bookmarking systems as a source of social information, but can be extended to consider other social sources such as comments, tweets, etc.

Through the LAICOS platform, the impact of our work in the industry becomes clearer. It remains the challenge of imposing such a platform either within the research community as an open source platform, or within the general public as a search engine.

8.2 Future Work

Besides the contributions presented above, short term and long term perspectives are still to be investigated. In the context of the problems tackled in this PhD thesis (problems related to social Web search), we envision some perspectives related to each of our contributions as follows:

— Improving the user's profile modeling in our algorithms. This is a perspective that we mentioned in Chapter 6.7. Indeed, the obtained results showed that modeling users' profiles with just the terms that they used is not always effective. The retrieval performance was decreasing for users with high profiles length. We believe that we can tackle this issue by considering in the users' profiles their social relatives [ACH+09], i.e. for a given user u, the list of users related to u, and the list of related terms to u.

— Taking into account the temporal dimension in our algorithms. Here, by temporal dimension we refer to two aspects:

  (i) Information related to the time when the query is issued. For example, if a user issues the query "restaurant" at 11:00pm, a response which include a closed restaurant at that time is irrelevant.

  (ii) Evolution of the tastes, interests and opinions of users. It is obvious that the user's mind may change over time for many reasons, e.g. he may grow, new information may impact his opinions, etc.

— Considering social relevance of entities in our algorithms. As discussed in Section 3.3.2.1, social relevance refers to information socially created that characterizes a document from a point of view of interest, i.e. its general interest, its popularity, etc. Basically, it can be interesting to consider the degree of trustworthiness between users, popularity of documents, trustiness of tags, co-authors relationships, etc., while processing a query. Such an information may make our algorithms more strong and effective regarding the results' quality.

— Improving the PSDR framework, where some weaknesses have been pointed in Chapter 5. In particular, we pointed the following two points:

  (i) Currently, we are investigating the possibility of using parallel computation to reduce the execution time of the PSDR framework. Indeed, the complexity analysis performed in Section 5.2.5 showed that the complexity of the PSDR framework scales linearly with the number of documents that match a query. The most likely track that we are investigating currently, is the use of the MapReduce programming model.

  (ii) We are also investigating ways to add social regularization terms to the objective function of Equation 5.6 in order to constrain it and model other social phenomenons, e.g. similarity between users, confidence of users, popularity of Web pages, etc.

— Evaluating our algorithms over other datasets, e.g. Flickr, CiteULike, Last.fm, Bibsonomy, etc. Currently, only the personalized social query expansion approach has been evaluated over three different datasets. However, this is not the case for the PSDR framework and SoPRa, which have been evaluated on only a *delicious* dataset. Evaluating on different datasets has two main purposes:

  (i) Confirming and consolidating the performance and the results obtained on different datasets.

  (ii) Studying and illustrating the algorithms' behavior on other topologies of social networks in general, and social tagging systems in particular. The algorithms may not behave similarly on different datasets as it is the case for the PSQE framework in Section 4.4.

— Formalizing a generalization algorithm to consider more types of social data. As pointed in Chapter 7, LAICOS is expected to leverage as different and heterogeneous source of social data as possible. We discussed a way to integrate these sources of social data, and to leverage them in our algorithms. However, currently, we didn't formalize and implement an algorithm for that.

Some long term perspectives are in the context of social recommendation. In particular, the topic that might interests our research team is to work on the temporal dimension of recommender systems to tackle the problems of: (i) hot topics, i.e. news, fresh information, (ii) the evolution of user profiles along time, i.e. the user interests evolve and change along time, and (iii) the diversity of information, i.e. in order not to annoy users with similar information. Indeed, for the first problem, information is time-dependent, meaning that it attracts much attention at a given moment and will be quickly forgotten after a while. We believe that the freshness of information is a key point while designing a social recommender system. The second problem deals with the evolution and the update of user profiles. For example, the opinion of a user concerning something may change in time when he grows, or reads news about this thing. The third problem is also a feeling that users have when they use Facebook in

particular. Most of the time, when a recent information appears, all Facebook friends of a particular user begin to publish articles that deal with the same information, and the user is quickly overwhelmed by similar information. We believe that, at a given time, the recommender system should know that the user is already aware about this information and consequently it should be hidden. These points have been mainly extracted from [NST+12].

Dealing with such issues is very interesting and motivating for our research team, even if we don't know how at the current moment. Do we have to deal with them by including social regularization terms in social matrix factorization to constrain the item features and user features? Or should we address these problems in another way, e.g. by enhancing other CF algorithms such as KNN or SVM? Or maybe we should propose new social recommendation algorithms? We believe that these problems should be deeply investigated.

Bibliography

[ABK+08] Sofiane Abbar, Mokrane Bouzeghoub, Dimitre Kostadinov, Stéphane Lopes, Armen Aghasaryan, and Stéphane Betge-Brezetz. A personalized access model: concepts and services for content delivery platforms. In *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '08, pages 41–47, New York, NY, USA, 2008. ACM.

[ABKL09] Sofiane Abbar, Mokrane Bouzeghoub, Dimitre Kostadinov, and Stéphane Lopes. A contextualization service for a personalized access model. In *9ème Conférence Francophone sur l'Extraction et la Gestion des Connaissances*, EGC '09, pages 265–270, 2009.

[ABL10] Sofiane Abbar, Mokrane Bouzeghoub, and Stéphane Lopes. Introducing contexts into personalized web applications. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '10, pages 155–162, New York, NY, USA, 2010. ACM.

[ACH+09] Einat Amitay, David Carmel, Nadav Har'El, Shila Ofek-Koifman, Aya Soffer, Sivan Yogev, and Nadav Golbandi. Social search and discovery using a unified approach. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, HT '09, pages 199–208, New York, NY, USA, 2009. ACM.

[AHK08] Fabian Abel, Nicola Henze, and Daniel Krause. Ranking in folksonomy systems: can context help? In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 1429–1430, New York, NY, USA, 2008. ACM.

[AYBLS08] Sihem Amer-Yahia, Michael Benedikt, Laks V. S. Lakshmanan, and Julia Stoyanovich. Efficient network aware search in collaborative tagging sites. *Proc. VLDB Endow.*, 1(1):710–721, August 2008.

[AYHY09] Sihem Amer-Yahia, Jian Huang, and Cong Yu. Building community-centric information exploration applications on social content sites. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 947–952, New York, NY, USA, 2009. ACM.

[AYLY09] Sihem Amer-Yahia, Laks V. S. Lakshmanan, and Cong Yu. SocialScope: Enabling Information Discovery on Social Content Sites. In *The biennial Conference on Innovative Data Systems Research*, CIDR '09, 2009.

[BBHB13] Mohamed Reda Bouadjenek, Amyn Bennamane, Hakim Hacid, and Mokrane Bouzeghoub. Evaluation of Personalized Social Ranking Functions of Information Retrieval. In Florian Daniel, Peter Dolog, and Qing Li, editors, *Web Engineering*, volume 7977 of *Lecture Notes in Computer Science*, pages 283–290. Springer Berlin Heidelberg, 2013.

[BCK+08] Matthias Bender, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Xavier Parreira, Ralf Schenkel, and Gerhard Weikum. Exploiting social relations for query expansion and result ranking. In *ICDE Workshops*, pages 501–506. IEEE Computer Society, 2008.

[Bel08] Nicholas J. Belkin. Some(what) grand challenges for information retrieval. *SIGIR Forum*, 42(1):47–54, June 2008.

[BFNP08] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 193–202, New York, NY, USA, 2008. ACM.

[BGLK09] Marin Bertier, Rachid Guerraoui, Vincent Leroy, and Anne-Marie Kermarrec. Toward personalized query expansion. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, SNS '09, pages 7–12, New York, NY, USA, 2009. ACM.

[BH12] Mohamed Reda Bouadjenek and Hakim Hacid. LAICOS: A Social Web Search Engine. In *World Wide Web CNRS Panel*, WWW '12, 2012.

[BHAC11a] Amyn Bennamane, Hakim Hacid, Arnaud Ansiaux, and Alain Cagnati. Aide à l'analyse visuelle de réseaux sociaux pour la detection de comportements suspects. In *11ème Conférence Francophone sur l'Extraction et la Gestion des Connaissances*, EGC '11, pages 221–226, 2011.

[BHAC11b] Amyn Bennamane, Hakim Hacid, Arnaud Ansiaux, and Alain Cagnati. Visual analysis of implicit social networks for suspicious behavior detection. In *Proceedings of the 16th international conference on Database systems for advanced applications: Part II*, DASFAA'11, pages 388–399, Berlin, Heidelberg, 2011. Springer-Verlag.

[BHB13a] Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. Laicos: an open source platform for personalized social web search. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '13, pages 1446–1449, New York, NY, USA, 2013. ACM.

[BHB13b] Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. Personalized Documents Ranking With Social Contextualization. In

10ème Colloque sur l'Optimisation et les Systèmes d'Information, COSI '13, pages 64–75, 2013.

[BHB13c] Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. SoPRa: A New Social Personalized Ranking Function for Improving Web Search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 861–864, New York, NY, USA, 2013. ACM.

[BHB16] Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. Social networks and information retrieval, how are they converging? a survey, a taxonomy and an analysis of social information retrieval approaches and platforms. *Information Systems*, 56:1 – 18, 2016.

[BHBD11a] Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Johann Daigremont. Personalized Social Query Expansion Using Social Bookmarking Systems. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1113–1114, New York, NY, USA, 2011. ACM.

[BHBD11b] Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Johann Daigremont. Une nouvelle approche d'expansion sociale de requêtes dans le web 2.0. In *8ème Conférence en Recherche d'Infomations et Applications*, CORIA '11, pages 41–48, 2011.

[BHBV13] Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Athena Vakali. Using Social Annotations to Enhance Document Representation for Personalized Search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 1049–1052, New York, NY, USA, 2013. ACM.

[BHJ+10] Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, and Gerd Stumme. Query Logs as Folksonomies. *Datenbank-Spektrum*, 10:15–24, 2010.

[BMS08] Claudio Biancalana, Alessandro Micarelli, and Claudio Squarcella. Nereau: a social approach to query expansion. In *Proceedings of the 10th ACM workshop on Web information and data management*, WIDM '08, pages 95–102, New York, NY, USA, 2008. ACM.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[Bro02] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, September 2002.

[BXW+07] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 501–510, New York, NY, USA, 2007. ACM.

[BYRN11] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley Longman Publishing Co., Inc., 2 edition, 2011.

[CGD+09] Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 201–210, New York, NY, USA, 2009. ACM.

[CRYT09] David Carmel, Haggai Roitman, and Elad Yom-Tov. Who tags the tags?: a framework for bookmark weighting. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1577–1580, New York, NY, USA, 2009. ACM.

[CRYT10] David Carmel, Haggai Roitman, and Elad Yom-Tov. Social bookmark weighting for search and recommendation. *The VLDB Journal*, 19:761–775, December 2010.

[CZ09] Shihn-Yuarn Chen and Yi Zhang. Improve Web Search Ranking with Social Tagging. *MSM*, 2009.

[CZG+09] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'el, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized social search based on the user's social network. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1227–1236, New York, NY, USA, 2009. ACM.

[DDR07] Arijit De, Elizabeth D. Diaz, and Vijay V. Raghavan. On Fuzzy Result Merging for Metasearch. In *FUZZ-IEEE'07*, pages 6–7, 2007.

[DEFS06] Pavel A. Dmitriev, Nadav Eiron, Marcus Fontoura, and Eugene Shekita. Using annotations in enterprise search. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 811–817, New York, NY, USA, 2006. ACM.

[DFDF04] Delbert Dueck, Brendan J. Frey, Delbert Dueck, and Brendan J. Frey. Probabilistic sparse matrix factorization. Technical report, 2004.

[DKMS11] Gideon Dror, Yehuda Koren, Yoelle Maarek, and Idan Szpektor. I want to answer; who has a question?: Yahoo! answers recommender systems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1109–1117, New York, NY, USA, 2011. ACM.

[DPK+12a] Sotiris Diplaris, Symeon Papadopoulos, Ioannis Kompatsiaris, Ayse Goker, Andrew Macfarlane, Jochen Spangenberg, Hakim Hacid, Linas Maknavicius, and Matthias Klusch. SocialSensor: sensing user generated input for improved media discovery and experience. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 243–246, New York, NY, USA, 2012. ACM.

[DPK+12b] Sotiris Diplaris, Symeon Papadopoulos, Ioannis Kompatsiaris, Nicolaus Heise, Jochen Spangenberg, Nic Newman, and Hakim Hacid. "making sense of it all": an attempt to aid journalists in analysing and filtering user generated content. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 1241–1246, New York, NY, USA, 2012. ACM.

[DZK+10] Anlei Dong, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 331–340, New York, NY, USA, 2010. ACM.

[Eft96] Efthimis N. Efthimiadis. Query expansion. *Annual Review of Information Systems and Technology (ARIST)*, 1996.

[FOdG10] Fernando Figueira Filho, Gary M. Olson, and Paulo Lício de Geus. Kolline: a task-oriented system for collaborative information seeking. In *Proceedings of the 28th ACM International Conference on Design of Communication*, SIGDOC '10, pages 89–94, New York, NY, USA, 2010. ACM.

[GE07] Georg Groh and Christian Ehmig. Recommendations in taste related domains: collaborative filtering vs. social filtering. In *Proceedings of the 2007 international ACM conference on Supporting group work*, GROUP '07, pages 127–136, New York, NY, USA, 2007. ACM.

[GF07] Dion Goh and Schubert Foo. *Social Information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2007.

[GGMP04] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 576–587. VLDB Endowment, 2004.

[GHS12] Kahina Gani, Hakim Hacid, and Ryan Skraba. Towards multiple identity detection in social networks. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 503–504, New York, NY, USA, 2012. ACM.

[GLYH10] Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. Survey on social tagging techniques. *SIGKDD Explor. Newsl.*, 12(1):58–72, November 2010.

[GZC+09] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 53–60, New York, NY, USA, 2009. ACM.

[GZR+10] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 194–201, New York, NY, USA, 2010. ACM.

[HBS10] John Hannon, Mike Bennett, and Barry Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 199–206, New York, NY, USA, 2010. ACM.

[HC09] Gary Hsieh and Scott Counts. mimir: a market-based real-time question and answer service. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 769–778, New York, NY, USA, 2009. ACM.

[HHHW08] C.C. Hung, Y.C. Huang, J.Y. Hsu, and D.K.C. Wu. Tag-Based User Profiling for Social Media Recommendation. In *Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at AAAI2008, Chicago, Illinois*, 2008.

[HHLS05] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools : A general review. *D-Lib Magazine*, 11(4), April 2005.

[HHM+11] Hakim Hacid, Karim Hebbar, Abderrahmane Maaradji, Mohamed Adel Saidi, Myriam Ribière, and Johann Daigremont. Enhancing navigation in virtual worlds through social networks analysis. In *Proceedings of the 19th international conference on Foundations of intelligent systems*, ISMIS'11, pages 146–152, Berlin, Heidelberg, 2011. Springer-Verlag.

[HJSS06a] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. FolkRank : A Ranking Algorithm for Folksonomies. In Klaus-Dieter Althoff and Martin Schaaf, editors, *LWA*, volume 1/2006 of *Hildesheimer Informatik-Berichte*, pages 111–114. University of Hildesheim, Institute of Computer Science, 2006.

[HJSS06b] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, pages 411–426, 2006.

[HK10] Damon Horowitz and Sepandar D. Kamvar. The anatomy of a large-scale social search engine. In *Proceedings of the 19th international conference*

on World wide web, WWW '10, pages 431–440, New York, NY, USA, 2010. ACM.

[HKGM08] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 195–206, New York, NY, USA, 2008. ACM.

[HMS11] John Hannon, Kevin McCarthy, and Barry Smyth. Finding useful users on twitter: twittomender the followee recommender. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 784–787, Berlin, Heidelberg, 2011. Springer-Verlag.

[JCC10] Bernard J. Jansen, Abdur Chowdury, and Geoff Cook. The ubiquitous and increasingly significant status message. *interactions*, 17(3):15–17, May 2010.

[JE10] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 135–142, New York, NY, USA, 2010. ACM.

[JLS09] Song Jin, Hongfei Lin, and Sui Su. Query expansion based on folksonomy tag co-occurrence analysis. In *2009 IEEE International Conference on Granular Computing*, 2009.

[JLW+11] Bernard J. Jansen, Zhe Liu, Courtney Weaver, Gerry Campbell, and Matthew Gregg. Real time search on the web: Queries, topics, and economic value. *Information Processing & Management*, 47(4):491 – 506, 2011.

[JMH+08] Robert Jäschke, Leandro Marinho, Andreas Hotho, Schmidt-Thie Lars, and Stum Gerd. Tag recommendations in social bookmarking systems. *AI Commun.*, 2008.

[KC09] Giridhar Kumaran and Vitor R. Carvalho. Reducing long queries using query quality predictors. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 564–571, New York, NY, USA, 2009. ACM.

[KFN09] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendations. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 61–68, New York, NY, USA, 2009. ACM.

[KHS08] Beate Krause, Andreas Hotho, and Gerd Stumme. A comparison of social bookmarking with traditional search. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 101–113, Berlin, Heidelberg, 2008. Springer-Verlag.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.

[LBM09] Christina Lioma, Roi Blanco, and Marie-Francine Moens. A logical inference approach to query expansion with social tags. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, ICTIR '09, pages 358–361, Berlin, Heidelberg, 2009. Springer-Verlag.

[LHHF05] Ben Lund, Tony Hammond, Timo Hannay, and Martin Flack. Social bookmarking tools (ii): A case study - connotea. *D-Lib Magazine*, 11(4), 2005.

[LL10] Fengkun Liu and Hong Joo Lee. Use of social network information to enhance collaborative filtering performance. *Expert Syst. Appl.*, 37:4772–4778, July 2010.

[LLJY11] Yuan Lin, Hongfei Lin, Song Jin, and Zheng Ye. Social annotation in query expansion: a machine learning approach. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 405–414, New York, NY, USA, 2011. ACM.

[LVHAM12] Juan Antonio Lossio Ventura, Hakim Hacid, Arnaud Ansiaux, and Maria Laura Maag. Conversations reconstruction in the social web. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 573–574, New York, NY, USA, 2012. ACM.

[MC05] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 472–479, New York, NY, USA, 2005. ACM.

[MCM+09] Benjamin Markines, Ciro Cattuto, Filippo Menczer, Dominik Benz, Andreas Hotho, and Gerd Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 641–650, New York, NY, USA, 2009. ACM.

[MH07] Meredith Ringel Morris and Eric Horvitz. Searchtogether: an interface for collaborative web search. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 3–12, New York, NY, USA, 2007. ACM.

[MHDC10] Abderrahmane Maaradji, Hakim Hacid, Johann Daigremont, and Noël Crespi. Towards a social network based approach for services composition. In *International Conference on Communications*, ICC '10, pages 1–5. IEEE, 2010.

[MHG10] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA, 2010.

[MHSV11] Abderrahmane Maaradji, Hakim Hacid, Ryan Skraba, and Athena Vakali. Social web mashups full completion via frequent sequence mining. In *World Congress on Services*, SERVICES '11, pages 9–16. IEEE Computer Society, 2011.

[Mik07a] P. Mika. *Social networks and the Semantic Web*. Semantic web and beyond. Springer, 2007.

[Mik07b] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semant.*, 5(1):5–15, 2007.

[Mis06] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 953–954, New York, NY, USA, 2006. ACM.

[MJSZ07] Qiaozhu Mei, Jing Jiang, Hang Su, and Chengxiang Zhai. Searching and tagging: Two sides of the same coin? Technical report, University of Illinois at UrbanaChampaign, 2007.

[MKK+08] Jayant Madhavan, David Ko, Lucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. Google's Deep Web crawl. *Proc. VLDB Endow.*, 1(2):1241–1252, August 2008.

[MKL09] Hao Ma, Irwin King, and Michael R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 203–210, New York, NY, USA, 2009. ACM.

[Mor07] Meredith Ringel Morris. Collaborating alone and together: Investigating persistent and multi-user web search activities. Technical report, Microsoft Research, 2007.

[Mor08] Meredith Ringel Morris. A survey of collaborative web search practices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1657–1660, New York, NY, USA, 2008. ACM.

[MTP10] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. What do People Ask Their Social Networks, and Why?: A Survey Study of Status Message Q&A Behavior. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1739–1748, New York, NY, USA, 2010. ACM.

[MYLK08] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 931–940, New York, NY, USA, 2008. ACM.

[MZL+11] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 287–296, New York, NY, USA, 2011. ACM.

[Nie06] Jakob Nielsen. Participation inequality: Encouraging more users to contribute, 2006.

[NM07] Michael G. Noll and Christoph Meinel. Web search personalization via social bookmarking and tagging. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 367–380, Berlin, Heidelberg, 2007. Springer-Verlag.

[NST+12] Joseph Noel, Scott Sanner, Khoi-Nguyen Tran, Peter Christen, Lexing Xie, Edwin V. Bonilla, Ehsan Abbasnejad, and Nicolas Della Penna. New objective functions for social collaborative filtering. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 859–868, New York, NY, USA, 2012. ACM.

[One09] The inner workings of a reatime search engine: Thoughts on realtime search, 2009.

[PM09] Sharoda A. Paul and Meredith Ringel Morris. Cosense: enhancing sensemaking for collaborative web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1771–1780, New York, NY, USA, 2009. ACM.

[Por97] M. F. Porter. *An algorithm for suffix stripping*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[PSC+02] James Pitkow, Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. Personalized search. *Commun. ACM*, 45(9):50–55, September 2002.

[RJ76] S. E. Robertson and K. Sparck Jones. Relevance Weighting of Search Terms. *J. Am. Soc. Inf. Sci.*, 27(3):129–146, 1976.

[RSJ88] Stephen E. Robertson and Karen Sparck Jones. Document retrieval systems. chapter Relevance weighting of search terms, pages 143–160. Taylor Graham Publishing, London, UK, UK, 1988.

[Sal68] Gerard. Salton. *Automatic Information Organization and Retrieval.* McGraw Hill Text, 1968.

[SAYMY08] Julia Stoyanovich, Sihem Amer-Yahia, Cameron Marlow, and Cong Yu. Leveraging tagging to model user interests in del.icio.us. In *AAAI Spring Symposium: Social Information Processing*, pages 104–109, 2008.

[SB02] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, New York, 3 edition, August 2002.

[SBC+10] Rossano Schifanella, Alain Barrat, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Folks in folksonomies: social link prediction from shared metadata. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 271–280, New York, NY, USA, 2010. ACM.

[SCK+08] Ralf Schenkel, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane X. Parreira, and Gerhard Weikum. Efficient top-k querying over social-tagging networks. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 523–530, New York, NY, USA, 2008. ACM.

[SK94] Wasserman Stanley and Faust Katherine. *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, 1 edition, November 1994.

[SM08] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[SNM10] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Trans. on Knowl. and Data Eng.*, 2010.

[SP99] Amit Singhal and Fernando Pereira. Document expansion for speech retrieval. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 34–41, New York, NY, USA, 1999. ACM.

[SS02] Munirathnam Srikanth and Rohini Srihari. Biterm language models for document retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 425–426, New York, NY, USA, 2002. ACM.

[TK08] Tsubasa Takahashi and Hiroyuki Kitagawa. S-bits: Social-bookmarking induced topic search. In *Proceedings of the 2008 The Ninth International Conference on Web-Age Information Management*, WAIM '08, pages 25–30, Washington, DC, USA, 2008. IEEE Computer Society.

[TK09] Tsubasa Takahashi and Hiroyuki Kitagawa. A ranking method for web search using social bookmarks. In *Proceedings of the 14th International Conference on Database Systems for Advanced Applications*, DASFAA '09, pages 585–589, Berlin, Heidelberg, 2009. Springer-Verlag.

[TRM11] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris. #twittersearch: a comparison of microblog search and web search. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 35–44, New York, NY, USA, 2011. ACM.

[VCJ10] David Vallet, Iván Cantador, and Joemon M. Jose. Personalizing web search with folksonomy-based user and document profiles. In *Proceedings of the 32nd European conference on Advances in Information Retrieval*, ECIR'2010, pages 420–431, Berlin, Heidelberg, 2010. Springer-Verlag.

[WHL11] Chen Wei, Wynne Hsu, and Mong Li Lee. A unified framework for recommendations based on quaternary semantic analysis. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1023–1032, New York, NY, USA, 2011. ACM.

[WJ10] Qihua Wang and Hongxia Jin. Exploring online social activities for adaptive search personalization. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 999–1008, New York, NY, USA, 2010. ACM.

[WLF11] Xufei Wang, Huan Liu, and Wei Fan. Connecting users with similar interests via tag network inference. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1019–1024, New York, NY, USA, 2011. ACM.

[WZB08] Robert Wetzker, Carsten Zimmermann, and Christian Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pages 26–30. IOS Press, 2008.

[XBF+08] Shengliang Xu, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 155–162, New York, NY, USA, 2008. ACM.

[YJNT07] Yusuke Yanbe, Adam Jatowt, Satoshi Nakamura, and Katsumi Tanaka. Towards improving web search by utilizing social bookmarks. In *Proceedings of the 7th international conference on Web engineering*, ICWE'07, pages 343–357, Berlin, Heidelberg, 2007. Springer-Verlag.

[YPL11] Le Yu, Rong Pan, and Zhangfeng Li. Adaptive social similarities for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 257–260, New York, NY, USA, 2011. ACM.

[YSL12] Xiwang Yang, Harald Steck, and Yong Liu. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1267–1275, New York, NY, USA, 2012. ACM.

[ZAAN07] Jun Zhang, Mark S. Ackerman, Lada Adamic, and Kevin Kyung Nam. Qume: a mechanism to support expertise finding in online help-seeking

communities. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 111–114, New York, NY, USA, 2007. ACM.

[ZYW+09] Xiaoxun Zhang, Lichun Yang, Xian Wu, Honglei Guo, Zhili Guo, Shenghua Bao, Yong Yu, and Zhong Su. sdoc: exploring social wisdom for document enhancement in web mining. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 395–404, New York, NY, USA, 2009. ACM.